National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES
THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

NL-339 (r. 86/06)

THE UNIVERSITY OF ALBERTA

# NUMERICAL AND EXPERIMENTAL STUDIES OF RIME ICE ACCRETION ON CYLINDERS AND AIRFOILS

by

© KAREN JOAN FINSTAD

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

IN

METEOROLOGY

DEPARTMENT OF GEOGRAPHY

EDMONTON, ALBERTA

Fall, 1986

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-32478-3

THE UNIVERSITY OF ALBERTA

**RELEASE FORM**

NAME OF AUTHOR KAREN JOAN FINSTAD

TITLE OF THESIS NUMERICAL AND EXPERIMENTAL STUDIES OF RIME ICE ACCRETION ON CYLINDERS AND AIRFOILS

DEGREE FOR WHICH THESIS WAS PRESENTED DOCTOR OF PHILOSOPHY

YEAR THIS DEGREE GRANTED Fall, 1986

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Karen J. Finstad

PERMANENT ADDRESS:

#901 - 10149 Saskatchewan Drive

Edmonton, Alberta

T6E 6B6

DATED July 9 1986

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled NUMERICAL AND EXPERIMENTAL STUDIES OF RIME ICE ACCRETION ON CYLINDERS AND AIRFOILS submitted by KAREN JOAN FINSTAD in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY in METEOROLOGY.

.....
Supervisor

.....

.....

.....

.....
External Examiner

Date..... July 9, 1986.....

## Abstract

Four separate studies describe improvements to numerical techniques used in the modelling of ice accretion on circular cylinders and on airfoils.

1. For collision efficiencies, accretion limits and impact velocities on circular cylinders, tables of numerical data and analytical approximating functions are presented. These data replace those of Langmuir and Blodgett (1946).
2. Parameterizations of the local collision efficiency distribution on circular cylinders are derived from numerical data, and applied in an 'operational' style computer model of rime ice accretion. The model's performance is compared with experimental accretions grown in the University of Alberta FROST tunnel.
3. The distribution of local ice density in rime ice is investigated in experimental wind-tunnel accretions made on circular cylinders. A form of the distribution is derived for application to the numerical model described in the previous study.
4. A second model, for rime ice accretion on airfoils or other arbitrary shapes, combines droplet trajectory integrations and parameterizations for local collision efficiency and density distribution. By use of these parameterizations, and of simple numerical methods throughout, the model is made significantly faster than previous models have been. Verification of the model is

provided by comparisons with several wind-tunnel experiments.

**Acknowledgements**

A most sincere thank-you is due to my thesis supervisor, Professor Edward Lozowski, for his expert guidance and perceptive advice, given throughout the course of my studies, and also for arranging generous research and conference travel funds . Professor Ted Gates, as co-leader of the Icing Group, was an equally valuable source of advice and expertise. To both of them, many thanks for always lending the time, and a good-natured ear, to my questions.

I would also like to express my appreciation to the members of my examination committee for their careful reading of the manuscript, and pertinent comments: Professor Erhard Reinelt, Assistant Professor John Wilson, and especially to Professor Gordon Rostoker for taking time from his busy schedule to serve as Chairman, and External Examiner Dr. Lasse Makkonen, for overcoming the difficulties presented by some 15,000 km and a Finnish postal strike.

Thanks also, to the other student members of the Icing Group, and to the students and staff of the Meteorology Division, for support and friendship.

Financial support in the form of Postgraduate Fellowships from the Atmospheric Environment Service, Canada, and a Graduate Faculty Fellowship from The University of Alberta, is gratefully acknowledged.

Finally, to J.A.H., my chief editor, programming advisor, and billiards marker - thanks for everything.

## Table of Contents


| Chapter | Page |
|---|---|
| I. Introduction | 1 |
| A. The Icing Problem | 3 |
| References | 9 |
| II. Cloud Droplet Trajectories Updated | 13 |
| A. Introduction | 13 |
| B. Impingement Parameters and Droplet Trajectories | 14 |
| C. Results | 20 |
| D. Curve Fitting | 23 |
| E. Limitations of LB Theory | 38 |
| F. Summary | 43 |
| References | 45 |
| III. An Operational Model for Rime Ice Accretion | 48 |
| A. Introduction | 48 |
| B. Parameterization for Local Collision Efficiency | 50 |
| C. Variation of Local Density | 55 |
| D. Rotation | 59 |
| E. Model Applications | 60 |
| References | 63 |
| IV. Local Densities of Circular Cylinder Rime Accretions | 65 |
| A. Introduction | 65 |
| B. Rotating Cylinder Densities Applied to the Non-rotating Case | 69 |
| C. Profile-Derived Density Distribution | 74 |
| D. Laboratory Measurements of Local Density | 77 |
| E. Local Density at the Stagnation Line | 86 |

F. Summary .....88
References .....90
V. Rime Icing on Airfoils .....92
A. Introduction .....92
B. Model Outline .....93
C. Specifying the Airfoil Shape .....95
D. Local Collision Efficiency .....99
E. Integration of Droplet Trajectories .....110
F. Local Density Variations .....115
G. Model Results .....116
References .....125
VI. Concluding Remarks .....128
Appendix I .....131
Appendix II .....146
Appendix III .....170
References .....174
Appendix IV .....175
Appendix V .....178
Appendix VI .....215
References .....228

## List of Tables


| Table | Page |
|---|---|
| II.1 - $\beta_0(K)$ for $\phi = 10^3$. | 21 |
| II.2 - Numerically calculated values of $\beta_0$, $\alpha_{max}$ and E as a function of K and $\phi$. | 24 |
| II.3 - Numerically calculated values of $V_0$ as a function of K and $\phi$. | 26 |
| II.4 - Approximations to the numerical data of Langmuir and Blodgett derived by various authors. | 31 |
| II.5 - Comparison of some original E values calculated by Langmuir and Blodgett, and those derived from the approximations of Table II.4. | 33 |
| II.6 - Coefficients for approximating impingement parameters by Equation II.11. | 37 |
| II.7 - Comparisons between a selection of numerically calculated values, and approximated values calculated according to Table II.6. | 39 |
| IV.1 - Local densities as a function of surface slope derived from thickness measurements of shallow ice layers. | 78 |
| IV.2 - Experimentally measured local densities as a function of surface slope. | 82 |

V.1 - Sensitivity of the integration results to the time step size and the initial x-position. 112

A.1 - Droplet size spectra from many different sources. 216

A.2 - Overall and stagnation line collision efficiencies as plotted in Figures A.2 and A.3. 221

A.3 - Overall and stagnation line collision efficiencies as plotted in Figures A.4 and A.5. 224

## List of Figures


| Figure | Page |
|---|---|
| II.1 - Droplet trajectories impinging on a circular cylinder. | 15 |
| II.2 - Comparison of $\beta_0$ vs. K for one value of $\phi = 10^3$. | 22 |
| II.3 - Smooth curves drawn for the numerical data of $\beta_0$ vs. K and $\phi$. | 27 |
| II.4 - Smooth curves drawn for the numerical data of $\alpha_{max}$ vs. K and $\phi$. | 28 |
| II.5 - Smooth curves drawn for the numerical data of E vs. K and $\phi$. | 29 |
| II.6 - Smooth curves drawn for the numerical data of $V_0$ vs. K and $\phi$. | 30 |
| II.7 - The dependence of $(K_{0,\beta}^{-}/K)$ on Reynolds Number Re. | 36 |
| II.8 - Potential flow streamlines about a cylinder. Viscous flow streamlines about a cylinder at a Reynolds number of 10,000. | 40 |
| III.1 - Parameterized $\beta(\alpha)$ values from Equations III.1 and III.2. | 53 |
| III.2 - Parameterized $\beta(\alpha)$ values from Equations III.3 and III.4. | 56 |
| III.3 - Comparisons between modelled and experimental profiles and masses per unit length. | 58 |

III.4 - An example of the model output for a rotating cylinder. 61

IV.1 - A 1.5 cm thick cross-section of a circular cylinder rime accretion, viewed in transmitted light. 66

IV.2 - A typical circular cylinder rime accretion showing rime feathers. 68

IV.3 - Experimental and three model accretions, showing the effects of three different density distributions. 72

IV.4 - A circular cylinder rime deposit in profile, with paint layers marking the surface at five minute intervals. 75

IV.5 - Local densities derived from thickness measurements of shallow ice layers. 79

IV.6 - An accretion section of about 1.5 cm thickness, showing the holes drilled for local density measurements. 81

IV.7 - Local density vs. surface angle; experimental measurements and parameterized curves. 84

IV.8 - Local density vs. surface angle; experimental measurements and parameterized curves. 85

IV.9 - Stagnation line density measurements vs. R', and the rotating cylinder density correlation of Makkonen and Stallabrass vs. R. 89

V.1 - Determination of collision effiency on an airfoil. 97

V.2 - Comparison of Oleskiw's numerical results with Equation V.3. 101

V.3 - Comparison of Oleskiw's numerical results with Equation V.3. 102

V.4 - Comparison of Oleskiw's numerical results with Equation V.3. 103

V.5 - Comparison of spectrum-weighted collision efficiencies and Equation V.5. 105

V.6 - Comparison of spectrum-weighted collision efficiencies and Equation V.5. 106

V.7 - Comparison of Oleskiw's numerical results with Equation V.7. 108

V.8 - Comparison of Oleskiw's numerical results with Equation V.7. 109

V.9 - Determination of droplet impact point on the surface. 114

V.10 - Comparison of present model results with those of Oleskiw. 118

V.11 - Comparison of present model results with an experimental accretion grown in the NASA/Lewis tunnel. 119

V.12 - Comparison of present model results with experimental profiles from Stallabrass and Stallabrass and Lozowski. 121

V.13 - Comparison of present model results with experimental profiles and masses grown in The University of Alberta FROST tunnel. 123

A.1 - Droplet trajectories impinging on an airfoil. 176

A.2 - Overall collision efficiencies calculated for many different drop size spectra vs. E's calculated for the mvd and d42 of each spectrum. 222

A.3 - Stagnation line collision efficiencies calculated for many different drop size spectra vs. $\beta_0$'s calculated for the mvd and d42 of each spectrum. 223

A.4 - Overall collision efficiencies calculated for a single drop size spectrum vs. E's calculated for the mvd and d42 of the same spectrum. 225

A.5 - Stagnation line collision efficiencies calculated for a single drop size spectrum vs. $\beta_0$'s calculated for the mvd and d42 of the same spectrum. 226

## List of Symbols

| Symbol | Definition |
|---|---|
| $C_d$ | - droplet drag coefficient |
| $C_{x,n}$ | - constant coefficients for Equation II.11 |
| $D_c$ | - circular cylinder diameter |
| $D_d$ | - droplet diameter |
| d42 | - diameter of droplet at the 42nd volume percentile of the droplet spectrum |
| E | - overall collision efficiency |
| $h_0$ | - airfoil width as a fraction of chord lengt |
| K | - Langmuir non-dimensional mass parameter |
| $K_0$ | - a function of K and Re, used in analytical fits to the data of Langmuir and Blodgett |
| lwc | - liquid water content |
| mvd | - median volume diameter droplet |
| R | - Macklin correlation parameter for rotating cylinder ice density |
| R' | - revised Macklin parameter for correlation of $\rho_0$ |
| $R_c$ | - circular cylinder diameter |
| Re | - droplet Reynolds number based on the free-stream velocity |
| $Re_d$ | - droplet Reynolds number based on the relative velocity |
| $r_d$ | - droplet radius |
| T | - temperature |

| | |
|---|---|
| Th | - thickness of ice layer |
| Tl | - tail length, added to $\alpha_{max}$ |
| $T_s$ | - mean surface temperature of ice layer |
| $T_{s,0}$ | - surface temperature of ice at the stagnation line |
| $\Delta t$ | - integration time step for droplet trajectories |
| U | - free-stream air speed |
| $u_x$, $u_y$ | - air speed coordinates |
| V | - relative speed of droplet with respect to the air |
| $V_r$ | - impact speed of droplet in the direction normal to the cylinder surface |
| $V_0$ | - droplet impact speed at the stagnation line |
| v | - absolute velocity of droplet |
| $v_x$, $v_y$ | - droplet speed coordinates |
| $x_0$, $y_0$ | - initial droplet coordinates |
| $\Delta y$ | - initial separation of trajectory pair |
| $\Delta \ell$ | - final separation in arc length of the trajectory pair |
| $\alpha$ | - angle between surface normal and the free-stream |
| $\alpha'$ | - scaled surface angle for Equation V.7 |
| $\alpha_0$ | - angular position of maximum $\beta$ |
| $\alpha_{max}$ | - angular position of accretion limit |
| $\alpha_m$ | - accretion limit plus tail length, Tl |

$\beta$ - local collision efficiency

$\beta_0$ - local collision efficiency at the stagnation line

$\zeta$ - Stallabrass correlation parameter

$\theta$ - position angle on circular cylinder

$\mu$ - dynamic viscosity of air

$\rho$ - local ice density

$\rho_a$ - density of air

$\rho_d$ - density of the water droplet

$\rho_0$ - local ice density at the stagnation line

$\tau$ - ice layer time step

$\Phi$ - angle of attack of the free-stream

$\phi$ - non-dimensional Langmuir flow parameter, or Stokes number

$\psi$ - potential flow streamfunction

# I. Introduction

'Icing' presents a problem occasionally to pastry cooks, sometimes to hockey players, and always to those concerned with the natural phenomenon of ice accretion.

In cold climates, ice accretion may result from atmospheric cloud droplets, rain, snow, or sea spray. It may cause serious hazards for aircraft, power lines and transmission towers, ships, and off-shore oil rigs. All of these represent human enterprises that involve many lives and dollars, which may be jeopardized during icing events. So, the primary aim in studying icing is the prediction of its occurrence, whether of a particular event or of the probability of extreme events, and ultimately the prevention of costly and often disastrous accidents.

The icing phenomenon may appear straightforward at first glance: water is collected on a cold surface and freezes there. However, a deeper look reveals many layers of complexity in the physical processes involved, and serious practical difficulties in the observational, experimental, and theoretical aspects of the problem. The development of a predictive model is therefore not an easy task, but it is nonetheless one of some importance.

As with other physical problems, there are two approaches that may be taken in model building. The first is a statistical approach, in which a very large number of events (predictands) are observed, as well as their antecedent or concurrent conditions (predictors). A

statistical analysis will then look for correlations between the predictors and predictands. Considerable computational effort may be required to establish a 'best guess' for the most significant predictors, and for the form of the predictive relation.

The reliability of the outcome is highly dependent on the number and quality of the observations. The advantage of this method is that a predictive relation may be derived and applied in the absence of any further understanding of the physical processes involved, which may be complex. But if the observations are adequate, and a good correlation can be found, it is the quickest way of achieving practical forecasting results.

However, the predictive skill of this method is limited by the strength of the discovered correlation, and its applicability is limited to the range of conditions defined by the original observations. For icing studies, this of course includes the geographical range, since the character of icing problems has a large regional variability.

The other approach to modelling is the physical one. This usually means a mathematical model, solved numerically using a digital computer, in which an attempt is made to simulate the intermediate steps, and to understand the causal relationships between the input conditions and the output predictions. Producing such a model does take considerably more time and effort than producing a statistical correlation, and the requirements for the

observational data are just as stringent. Furthermore, no mathematical model can perfectly reproduce a given physical process or situation - there must always be simplifying assumptions.

But the knowledge gained from a physical model is less restricted in its application, within those assumptions, than is any purely statistical relationship, and it can provide a useful testing ground for theory. Even the simplest physical model might at least suggest what the most effective predictors will be for a statistical study; thus if it is found that liquid water content and cloud droplet size are of prime importance in determining model ice accretion rates, then collectors of field data should take note, even if these things are very difficult to observe.

**A. The Icing Problem**

A model of the complete ice accretion process involves the solution of a series of problems. First, an approximation to the airflow around the accreting object must be found. If potential flow is assumed, and there is no analytical solution (as there is for circular cylinders or Joukowski airfoils) then the partial differential equation for the velocity potential must be solved numerically, as in Hess and Smith (1967), or Kennedy and Marsden (1976), for example.

Once it is possible to calculate the air velocity at any point, the trajectories of water droplets carried by the

airstream toward the object may be determined. Again, simplifying assumptions are applied to the droplet's equation of motion in order to reduce the computational time needed to integrate it numerically.

An ensemble of droplet trajectories is calculated, and their points of impact on the accretion surface determined. This information is necessary to derive the impingement characteristics of the accretion shape. These are: the distribution of the local collision efficiency, the overall collision efficiency, the maximum impingement angle, and impact speed of the droplets. The collection efficiency is the product of the collision efficiency and the sticking efficiency. Since the latter is unknown, but usually assumed to be unity, only the collision efficiency is dealt with. These quantities and their symbols are defined for a circular cylinder in Figure II.1.

Langmuir and Blodgett (1946) authored the classical solution to this problem, for collision efficiencies of cylinders, spheres, and ribbons in a potential flow field. More recent, but less extensive calculations have been made by McComber and Touzot (1981), and by Egelhofer, et al. (1984). The experimental work of Ranz and Wong (1952) qualitatively validates the theory of Langmuir and Blodgett, if not their numerical calculations.

The impingement characteristics depend on the cylinder size, droplet size, the free-stream air speed, and the air temperature and pressure. If more than one droplet size is

present, the impingement characteristics of each size, or size interval, must be determined. A common simplification to this requirement for droplet spectra is made through the substitution of the impingement characteristics of a single droplet size, that of the median volume droplet of the size spectrum. The usefulness of this simplification is examined in Appendix VI, although it is used throughout the thesis.

Only when all of these calculations are completed can ice begin to accrete in the model. For obvious reasons then, most modelers have turned to standard accretion shapes for which these calculations are simpler, for which much data already exist, and for ease of comparison between models. Thus we have the ubiquitous horizontal circular cylinder in a uniform cross-flow, which appears in the models of Macklin and Payne (1967), McComber and Touzot (1981), Lozowski, Stallabrass and Hearty (1983), Horjen (1983), Egelhofer, et al. (1984) and Makkonen (1984). The more difficult problem of accretion on airfoils has been treated by Oleskiw (1982), Gent and Cansdale (1985), and MacArthur, et al. (1982).

The cloud water is usually assumed to be at the temperature of the air stream, and to arrive at the accreting surface in a uniform flux determined by the liquid water content, or lwc, (defined as the mass of water contained in a unit volume of air), the air speed and the object's collision efficiency. The density of the accreted ice also determines the thickness and shape of the deposit.

On rotating cylinders, accreted rime ice densities have been examined by Macklin (1962), and by Makkonen and Stallabrass (1985). On non-rotating objects, the shape of the deposit is also partly dependent on the variation of the local ice density along its surface. Bain and Gayet (1982) have suggested one form for this variation.

If the flux of super-cooled water is large enough so that the latent heat of freezing cannot be radiated, conducted or convected away from the surface fast enough, then the surface temperature may rise to the melting point. When this happens the ice may become 'wet', rather than 'dry', meaning that all of the impinging water is no longer able to freeze on impact. Now the problem becomes even more complex since a detailed heat balance must be calculated for the accretion surface, in order to determine how much of the impinging water is able to freeze, and how much is shed, or remains in the accretion as liquid water. Calculation of the heat balance is made especially difficult since the heat transfer properties of the ice surfaces and even of rough circular cylinders are very poorly known (Makkonen, 1985).

Models in which a computation of the heat balance on cylinders is attempted using empirical heat transfer data include those of Lozowski, Stallabrass and Hearty (1984), Macklin and Payne (1967), Egelhofer, et al. (1984), and Makkonen (1984). However, such data are scarce, especially for non-cylindrical shapes, and for rough surfaces. Some modelers have thus used numerically calculated heat transfer

for cylinders (Makkonen, 1985) or for airfoils (Cansdale and McNaughton, 1977, Cansdale and Gent 1983).

Models developed specifically for marine applications must contend with the further complications of salinity, droplet distributions and liquid water contents that vary with height, and intermittent sprays. Models including these effects are currently under development by various researchers.

The models developed for this thesis are intended as 'operational' icing models, which would be efficient and inexpensive enough to be used regularly in the laboratory for predictive purposes, or for numerical experiments into specific aspects of icing. Previous models have often been much too costly to be used in such a way. To reduce computational time several 'shortcuts', or parameterizations, have been introduced which help to simplify some of the calculations needed in this problem. Specifically, the local collision efficiency and local ice density are approximated by analytical functions of the surface slope. Such an approach naturally involves some loss of accuracy, but this loss must be balanced against unavoidable uncertainties in the input data and the required precision of output predictions.

Two models are described, for circular cylinders in Chapter III, and in Chapter V for airfoils, that could also be applied to other arbitrary shapes. Both models are limited to rime ice accretion, i.e. there is no heat balance

calculation, and a uniform air speed and flux of pure water droplets are assumed. This is so that the performance of the parameterizations can be easily compared with real ice accretions whose shape and mass are not dependent on the poorly known heat transfer distribution.

In the course of model development, significant improvements were made in two specific aspects, the theoretical calculation of collision efficiencies on smooth circular cylinders, (Chapter II) and the distribution of local density in non-rotating circular cylinder rime accretions (Chapter IV).

Experiments for model verification were carried out in the University of Alberta's original icing wind-tunnel, known as the Facility for Research On Solidification and Thawing, or FROST tunnel. A full description of the facility may be found in Gates (1981), or Lozowski and Gates (1984). All wind-tunnel experiments were performed under the direction of Professor Gates.

## References


Bain, M. and Gayet, J-F. 1982: Contribution to the Modelling of the Ice Accretion Process: Ice Density Variation with the Impacted Surface Angle. Proceedings, First International Workshop on Atmospheric Icing of Structures, Hanover, 13-20.

Cansdale, J.T. and Gent, R.W. 1983: Ice Accretion on Aerofoils in Two-Dimensional Compressible Flow - A Theoretical Model. British Royal Aircraft Establishment Technical Report 82128, 31 pp. + 31 fig.s.

Cansdale, J.T. and McNaughtan, I.I. 1977: Calculation of Surface Temperature and Ice Accretion Rate in a Mixed Water Droplet / Ice Crystal Cloud. British Royal Aircraft Establishment, Technical Report 77090, 24 pp. + 6 fig.s.

Egelhofer, K.Z., Ackley, S.F. and Lynch, D.R. 1984: Computer Modelling of Atmospheric Ice Accretion and Aerodynamic Ice Loading of Transmission Lines. Proceedings, Second International Workshop on Atmospheric Icing of Structures, Trondheim, (in press).

Gates, E.M. 1981: FROST Tunnel. Departmental Report No. 26, Department of Mechanical Engineering, The University of

Alberta. 10 pp.

Gent, R.W. and Cansdale, J.T. 1985: The Development of Mathematical Modelling Techniques for Helicopter Rotor Icing. AIAA Publication 85-0336, 10 pp.

Hess, J.L. and Smith, A.M.O. 1967: Calculation of Potential Flow About Arbitrary Bodies, *Progress in Aeronautical Sciences, Vol. 8*, Pergamon Press, 1-138.

Horjen, I. 1983: Icing on Offshore Structures Atmospheric Icing. *Norwegian Maritime Research*, II, No. 3, 9-21.

Kennedy, J.L. and Marsden D.J. 1976: Potential Flow Velocity Distributions on Multi-Component Airfoil Sections. *Can.Aero. and Space Jour.*, 22, 243-256.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories. Collected Works of Irving Langmuir, Vol.10, Pergamon Press, 348-393.

Lozowski, E.P. and Gates, E.M. 1984: Ice Accretion in a Marine Environment. Proceedings, 3rd International Symposium on Offshore Mechanics and Arctic Engineering, New Orleans, 8 pp.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Non-rotating Cylinder Part I: A Simulation Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

MacArthur, C.D. Keller, J.L. and Luers, J.K. 1982: Mathematical Modelling of Ice Accretion on Airfoils. AIAA Publication No. 82-0284.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Cylinder by the Finite Element Method. *J. Atmos. Sci.*, 38, 1027-1036.

Macklin, W.C. 1962: The Density and Structure of Ice Formed by Accretion. *Quart. Jour. Roy. Meteor. Soc.* 88, 30-50.

Macklin, W.C. and Payne, G.S. 1967: A Theoretical Study of the Ice Accretion Process. *Quart. Jour. Roy. Meteor. Soc.*, 93, 195-213.

Makkonen, L. 1984: Modelling of Ice Accretion on Wires. *Jour. Climate and App. Meteor.*, 23, 929-939.

Makkonen, L. 1985: Heat Transfer and Icing of a Rough Cylinder. *Cold Regions Science and Technology*, 10, 105-116.

Makkonen, L. and Stallabrass, J.R. 1984: Ice Accretion on Cylinders and Wires. National Research Council, Mechanical Engineering Division, Technical Report TR-LT-005, 50 pp.

Poots, G. 1984: Theoretical Solutions of Ice Accretion on Cables. Proceedings, Second International Workshop on Atmospheric Icing of Structures, (in press).

Ranz, W.E. and Wong, J.B. 1952: Impaction of Dust and Smoke Particles. *Industrial Engineering Chemistry*, 44, 1371-1381.

## II. Cloud Droplet Trajectories Updated

### A. Introduction

For many years the standard reference work for cloud droplet trajectories in an airstream flowing around an obstacle has been Langmuir and Blodgett (1946), hereafter LB. This was one of the first published treatments of the problem which used an analog computer to integrate the equations of motion for water droplets in a steady potential flow about an infinitely long circular cylinder.

These calculations are of interest in studies of atmospheric and marine ice accretion, in which the characteristics of droplet impingement on cylinders and other objects are important factors in modelling the icing process. However, since the calculations were done so long ago, on what would now be considered a primitive machine, and because the original reference can be difficult to obtain, a need was perceived to recalculate the droplet trajectories and to publish the results in the open literature. The effect of replacing the drag coefficient formulation used by LB with a more recent one, that of Beard and Pruppacher (1969), has also been investigated.

Functional fits have been devised by several authors to the original tabulated data of LB. A number of these are compared to show the need for improving and standardizing them. Toward this end, a non-linear regression fit to the new data has been found for the range of conditions

pertaining to airborne, land structure and marine icing.

## B. Impingement Parameters and Droplet Trajectories

From trajectory calculations may be derived the overall collision efficiency, E, the local collision efficiency at a point on the surface represented by its local surface slope, $\beta(\alpha)$, the local collision efficiency at the stagnation line, $\beta_0$, the maximum impingement angle on the cylinder surface, $\alpha_{max}$, and the non-dimensional impact velocity at the stagnation line, $V_0$. $V_0$ is normalized by the free stream speed, U. The local surface angle, $\alpha$, is defined as the angle between the local surface normal and the free stream direction. The other quantities are defined in Figure II.1.

Values of E, $\beta_0$, $\alpha_{max}$ and $V_0$ are given by LB for a large number of cases which vary in droplet size, cylinder size, and air speed. These results are tabulated in terms of the non-dimensional parameters, K:

$$K = (\rho_d D_d^2 U) / (9 \mu D_c), \qquad \text{II.1}$$

and $\phi$:

$$\phi = Re^2 / K, \qquad \text{II.2}$$

with the droplet Reynolds number based on the free-stream velocity, Re, given by:

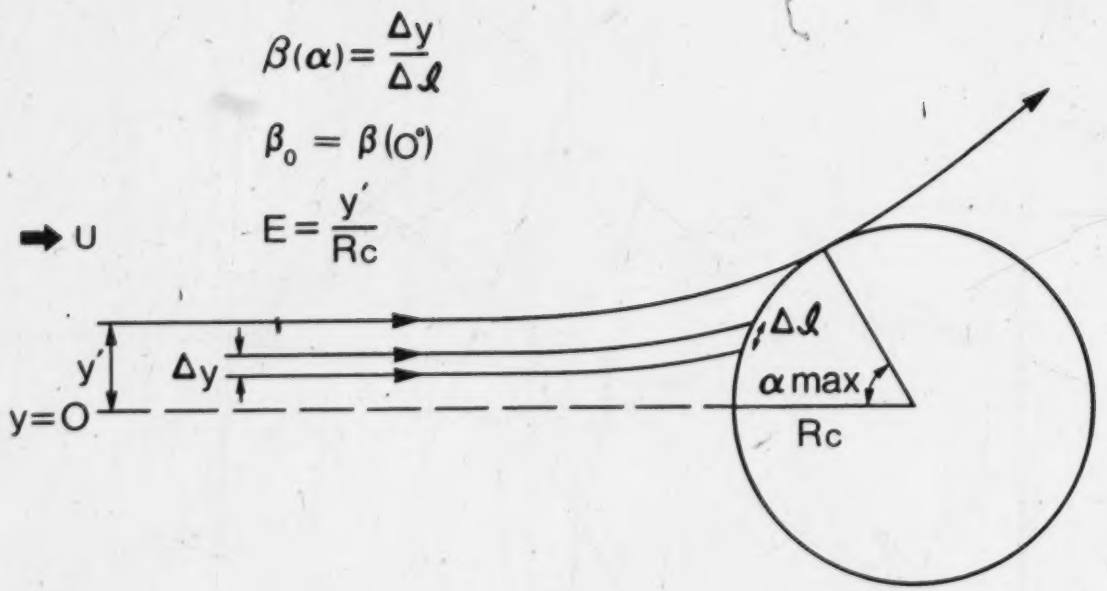



Figure II.1 - Droplet trajectories impinging on a circular cylinder, giving the definitions of $\beta(\alpha)$, $\beta_0$, E and $\alpha_{max}$. On the accretion surface $\alpha$ is the angle between the local surface normal and the free stream direction.

$$Re = U D_d \rho_a / \mu. \qquad II.3$$

Here $\rho_d$ is the density of water, $\rho_a$ the density of air, $\mu$ the dynamic viscosity of air, $D_d$ the droplet diameter, $D_c$ the cylinder diameter, and U the free stream air speed.

The K parameter measures the inertia of the droplet, and is also known as the Stokes number. To aid in solving problems in which the droplet size is the unknown quantity, LB presented their data in terms of a second non-dimensional parameter , $\phi$, that is independent of $D_d$.

The case $\phi = 0$ corresponds to Stokes, or creeping flow:

$$C_d Re_d / 24 = 1, \qquad II.4$$

where $C_d$ is the droplet drag coefficient, and $Re_d$ is the droplet Reynolds number based on the relative velocity with respect to the air, V, so that:

$$Re_d = |V| Re / U. \qquad II.5$$

In component form, the droplet equations of motion are:

$$dv_x / dt = -(v_x - u_x) (U C_d Re_d) / (12 D_c K)$$

$$dv_y / dt = -(v_y - u_y) (U C_d Re_d) / (12 D_c K), \qquad II.6$$

where t is time, $v_x$, $v_y$ and $u_x$, $u_y$ are the droplet and air

speed components, respectively. This set of equations is in dimensional form, with the origin at the cylinder axis and U in the positive x direction. Gravity and the effects of droplet acceleration on the drag have been neglected, as in the original theory of LB.

The air speed components for potential flow around a cylinder (Green, 1937) are:

$$u_x(x,y) = U\,(1 + (R_c^2\,(y^2 - x^2) / (x^2 + y^2)^2))$$

$$u_y(x,y) = -(2\,U\,R_c^2)\,(x\,y) / (x^2 + y^2)^2. \qquad \text{II.7}$$

In order to permit a direct comparison with the original solutions by LB some trajectories have been integrated using their empirical formulation for the steady-state drag coefficient:

$$C_d\,Re_d / 24 = 1 + 0.197\,Re_d^{0.63} + 2.6*10^{-4}\,Re_d^{1.38} \qquad \text{II.8}$$

More recent studies have improved the accuracy of $C_d$ approximations at low Reynolds numbers, however, so in further calculations over the complete range of conditions the formulation of Beard and Pruppacher (1969) has been substituted when $Re_d$ is less than 200. This formulation is:

$$C_d\ Re_d/24 = 1 + 0.102\ Re_d^{0.955},\ \text{for } 0.2 \le Re_d \le 2.0$$

$$C_d\ Re_d/24 = 1 + 0.115\ Re_d^{0.802},\ \text{for } 2.0 \le Re_d \le 21.0 \qquad \text{II.9}$$

$$C_d\ Re_d/24 = 1 + 0.189\ Re_d^{0.632},\ \text{for } 21.0 \le Re_d \le 200.0$$

A comparison of a variety of theoretical and experimental results for the drag coefficients of spheres is given by Beard (1976), in which the above formulation is shown to be an accurate analytical approximation.

The integration is carried out by the Heun method as described in Mesinger and Arakawa (1976). The droplet trajectories are begun at ten cylinder radii upstream of the origin with speeds at the initial position $(x_0,y_0)$ assumed to be:

$$v_x(x_0,y_0) = u_x(x_0,y_0) \qquad \text{II.10}$$

$$v_y(x_0,y_0) = 0.5\ u_y(x_0,y_0)$$

At this initial upstream distance, the final results are not greatly influenced by the choice of initial droplet speed as long as it lies in between the local air speed and the undisturbed upstream air speed. Tests indicate differences in collision efficiency of less than 0.5 percent, when the initial droplet speeds are varied within this range.

Other factors which influence the accuracy of the integration are the time step, $\Delta t$, and the initial upstream distance, $x_0$. A value of $x_0$ = 10 cylinder radii has been chosen, which, when compared to the results with $x_0$ = 20 cylinder radii, agrees to within 0.7 percent. At this initial distance, the size of the time step made less than a 0.5 percent difference in the results, as long as $\Delta t < 0.0033\ (D_c/U)$. The value used is $0.0025\ (D_c/U)$. Double precision is used throughout the program.

Values of the air viscosity and air density used are based on an assumed air temperature of -10° C. Within the temperature range of interest for most dry icing, i.e. -20 to 0° C, the collision efficiencies vary by a maximum of ±1 percent.

An air pressure of 100 kPa was assumed. The calculations of LB were carried out for an atmospheric pressure of 78.5 kPa, for application to Mt. Washington, or to airborne icing in general. At this pressure value, the present results differed from the 100 kPa results by less than 0.5 percent. The new results should therefore be applicable to both ground-level and airborne icing.

Two complete trajectories must be integrated in order to calculate one value of the local collision efficiency $\beta(\alpha)$, where $\alpha$ is the mean angle of the two impact points. As illustrated in Figure II.1, $\beta(\alpha)$ is defined as the ratio of the initial vertical separation of the two trajectories, to the final separation in arc length of the two impact

points. In the present model, the initial y separation of the two trajectories is always $10^{-8}$ metres, an arbitrarily small distance for use in the finite difference approximation to $dy/d\ell$, the definition of local collision efficiency. The stagnation line value, $\beta_0$, is calculated from the case where the first trajectory's initial position $y_0 = 0$.

The maximum impingement angle, $\alpha_{max}$, is found from the droplet trajectory which is tangent to the cylinder surface at its point of impact. This 'grazing' trajectory is found by increasing the initial position $y_0$ by successively smaller amounts until the tangent trajectory is reached.

The total collision efficiency E may then be calculated from the ratio of the initial vertical distance $y_0$(graze) to the cylinder radius.

The Fortran77 code written for these calculations is included in Appendix I.

## C. Results

In Figure II.2 three curves of $\beta_0$ vs. K are shown for the value $\phi = 10^3$, to illustrate the differences between LB's original calculations, the re-calculation of their results, and the re-calculation using Beard and Pruppacher's formulation for drag coefficient. The data for this Figure are listed in Table II.1.

A comparison between the first two curves shows that the LB results are too large for small K ($K \leq 0.4$), by up to

| K | $\beta_o$ (LB), old $C_d$ | $\beta_o$ (new), old $C_d$ | $\beta_o$ (new), new $C_d$ |
|---|---|---|---|
| .144 | 0.048 | 0.007 | 0.011 |
| .196 | 0.095 | 0.058 | 0.073 |
| .256 | 0.146 | 0.116 | 0.137 |
| .4 | 0.233 | 0.224 | 0.247 |
| .625 | 0.323 | 0.335 | 0.359 |
| .9 | 0.398 | 0.425 | 0.447 |
| 1.6 | 0.513 | 0.558 | 0.575 |
| 3.6 | 0.680 | 0.714 | 0.724 |
| 6.4 | 0.778 | 0.798 | 0.806 |
| 10. | 0.837 | 0.848 | 0.854 |
| 19.6 | 0.901 | 0.902 | 0.906 |
| 32.4 | 0.932 | 0.929 | 0.932 |
| 62.5 | 0.959 | 0.953 | 0.955 |
| 90. | 0.969 | 0.962 | 0.963 |
| 160. | 0.980 | 0.972 | 0.973 |
| 360. | 0.989 | 0.981 | 0.981 |
| 640. | 0.993 | 0.984 | 0.984 |
| 1000. | 0.995 | 0.986 | 0.986 |

**Table II.1 - $\beta_0(K)$ for $\phi = 10^3$. A comparison between Langmuir and Blodgett's results and the new results.**

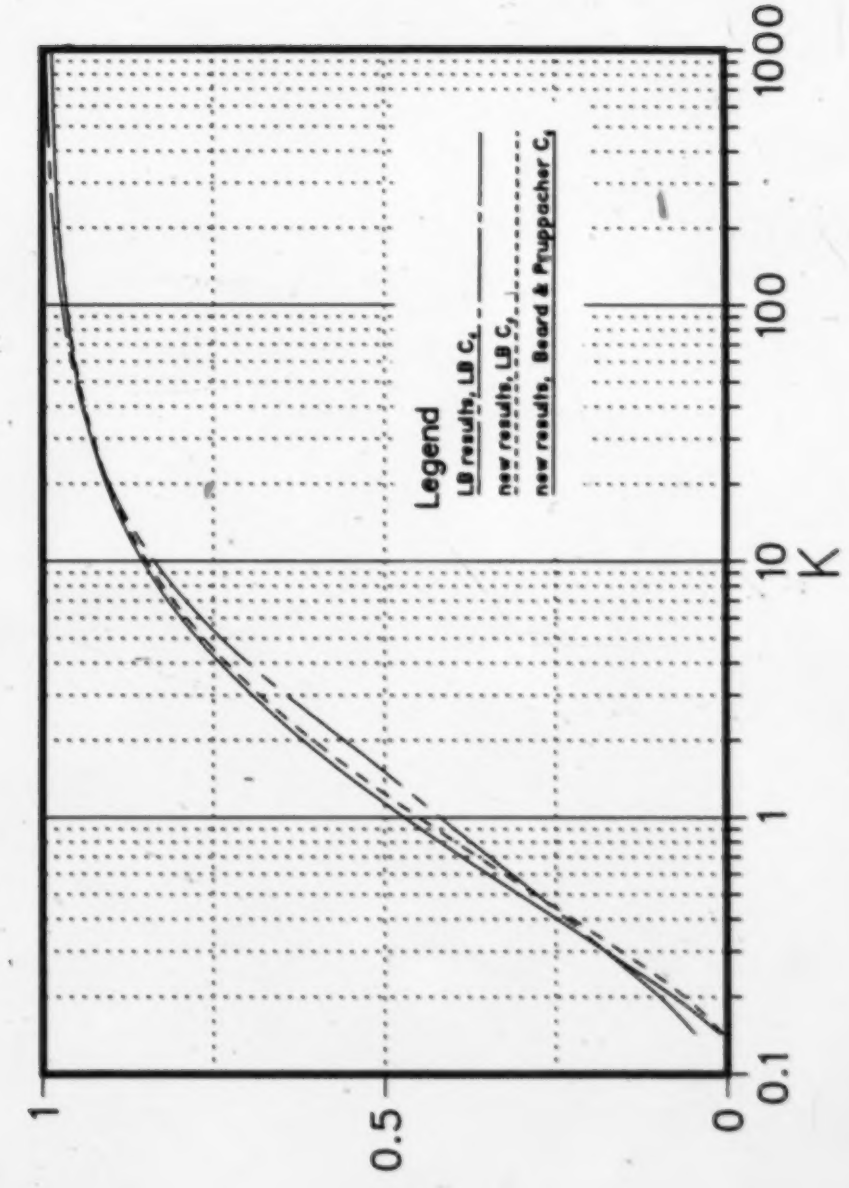



Figure II.2 - Comparison of $\beta_0$ vs. K for one value of $\phi = 10^3$.

several hundred percent. In the range $0.4 < K < 20.0$ their results are too small, by amounts near 10 percent in the middle of the range. For larger K, their values are again larger than the new results, but only by a few percent. The use of the revised drag coefficients (dotted curve, Figure II.2) gives another small increase over the LB results for all $K < 200.0$. A noticeable difference is that the new results for $\beta_0$ reach an effective value of zero for $K \simeq 0.14$, rather than at LB's theoretical value of 0.125. This difference is as yet unexplained. Results for other $\phi$ values, and for $\alpha_{max}$, E and $V_0$ are quantitatively similar in differences between the old and new results.

It is concluded from the comparisons in Figure II.2 and Table II.1, that the original LB calculations are accurate only for very large K, but that when K is less than about 200 they may be significantly in error. The new results are suggested for future use. These are shown for $\beta_0$, $\alpha_{max}$, E and $V_0$ for a complete range of $\phi$ and K in Tables II.2 to II.3, and in Figures II.3 through II.6, where the tabulated data have been fit by spline curves.

**D. Curve Fitting**

LB also drew smooth curves through their data, and derived analytical expressions in K and $\phi$ to fit them. Subsequently other authors have made attempts to improve on or to simplify these relations, and have then applied them

| K | $\varphi = 0$ | | | $\varphi = 10$ | | | $\varphi = 10^3$ | | | $\varphi = 10^5$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\beta_o$ | $a_{max}$ | E | $\beta_o$ | $a_{max}$ | E | $\beta_o$ | $a_{max}$ | E | $\beta_o$ | $a_{max}$ | E |
| .144 | 0.014 | 0.018 | 0.0002 | 0.013 | 0.009 | 0.0001 | 0.011 | 0.015 | 0.0001 | 0.004 | 0.007 | 0.00002 |
| .196 | 0.094 | 0.170 | 0.012 | 0.091 | 0.159 | 0.011 | 0.073 | 0.131 | 0.007 | 0.030 | 0.054 | 0.001 |
| .256 | 0.175 | 0.328 | 0.043 | 0.170 | 0.310 | 0.040 | 0.137 | 0.247 | 0.025 | 0.058 | 0.105 | 0.004 |
| .4 | 0.313 | 0.580 | 0.131 | 0.304 | 0.553 | 0.122 | 0.247 | 0.436 | 0.079 | 0.113 | 0.195 | 0.016 |
| .625 | 0.446 | 0.799 | 0.249 | 0.432 | 0.760 | 0.233 | 0.359 | 0.612 | 0.157 | 0.179 | 0.302 | 0.039 |
| .9 | 0.545 | 0.953 | 0.355 | 0.530 | 0.905 | 0.333 | 0.447 | 0.742 | 0.234 | 0.244 | 0.400 | 0.070 |
| 1.6 | 0.680 | 1.150 | 0.521 | 0.663 | 1.095 | 0.492 | 0.575 | 0.921 | 0.365 | 0.353 | 0.559 | 0.142 |
| 3.6 | 0.821 | 1.345 | 0.718 | 0.805 | 1.298 | 0.686 | 0.725 | 1.136 | 0.553 | 0.508 | 0.764 | 0.274 |
| 6.4 | 0.886 | 1.433 | 0.818 | 0.872 | 1.383 | 0.789 | 0.806 | 1.255 | 0.670 | 0.601 | 0.898 | 0.376 |
| 10. | 0.921 | 1.476 | 0.873 | 0.909 | 1.439 | 0.848 | 0.854 | 1.329 | 0.745 | 0.670 | 0.997 | 0.459 |
| 19.6 | 0.954 | 1.521 | 0.927 | 0.945 | 1.489 | 0.908 | 0.906 | 1.413 | 0.832 | 0.759 | 1.130 | 0.580 |
| 32.4 | 0.968 | 1.538 | 0.951 | 0.961 | 1.513 | 0.936 | 0.932 | 1.457 | 0.878 | 0.812 | 1.214 | 0.661 |
| 62.5 | 0.979 | 1.552 | 0.970 | 0.974 | 1.532 | 0.959 | 0.955 | 1.496 | 0.920 | 0.866 | 1.303 | 0.749 |
| 90. | 0.983 | 1.556 | 0.976 | 0.979 | 1.539 | 0.967 | 0.963 | 1.512 | 0.935 | 0.889 | 1.347 | 0.789 |
| 160. | 0.986 | 1.563 | 0.982 | 0.984 | 1.545 | 0.976 | 0.973 | 1.530 | 0.953 | 0.917 | 1.400 | 0.840 |
| 360. | 0.989 | 1.567 | 0.987 | 0.987 | 1.553 | 0.983 | 0.981 | 1.546 | 0.969 | 0.945 | 1.454 | 0.892 |
| 640. | 0.990 | 1.568 | 0.988 | 0.989 | 1.559 | 0.986 | 0.984 | 1.553 | 0.976 | 0.957 | 1.481 | 0.917 |
| 1000. | 0.990 | 1.568 | 0.989 | 0.989 | 1.557 | 0.987 | 0.986 | 1.557 | 0.980 | 0.965 | 1.499 | 0.932 |

Table II.2 - Numerically calculated values of $\beta_0$, $\alpha_{max}$ and E as a function of K and $\phi$.

| K | $\varphi = 10^2$ | | | $\varphi = 10^4$ | | | $\varphi = 10^6$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_o$ | $a_{max}$ | E | $\beta_o$ | $a_{max}$ | E | $\beta_o$ | $a_{max}$ | E |
| .16 | 0.034 | 0.054 | 0.001 | 0.021 | 0.035 | 0.0006 | 0.005 | 0.011 | 0.00004 |
| .25 | 0.153 | 0.282 | 0.032 | 0.094 | 0.168 | 0.012 | 0.026 | 0.051 | 0.001 |
| .36 | 0.257 | 0.466 | 0.088 | 0.164 | 0.287 | 0.034 | 0.050 | 0.089 | 0.003 |
| .64 | 0.415 | 0.720 | 0.212 | 0.281 | 0.468 | 0.095 | 0.103 | 0.176 | 0.013 |
| 1. | 0.528 | 0.889 | 0.325 | 0.374 | 0.609 | 0.163 | 0.157 | 0.260 | 0.029 |
| 1.96 | 0.676 | 1.097 | 0.500 | 0.518 | 0.817 | 0.296 | 0.251 | 0.380 | 0.067 |
| 3.24 | 0.763 | 1.219 | 0.617 | 0.617 | 0.958 | 0.406 | 0.319 | 0.488 | 0.112 |
| 6.25 | 0.848 | 1.336 | 0.743 | 0.726 | 1.115 | 0.547 | 0.424 | 0.633 | 0.191 |
| 9. | 0.882 | 1.387 | 0.797 | 0.777 | 1.187 | 0.618 | 0.482 | 0.712 | 0.243 |
| 16. | 0.921 | 1.446 | 0.863 | 0.838 | 1.278 | 0.711 | 0.570 | 0.832 | 0.333 |
| 36. | 0.955 | 1.501 | 0.922 | 0.899 | 1.379 | 0.812 | 0.680 | 0.982 | 0.463 |
| 64. | 0.968 | 1.523 | 0.946 | 0.927 | 1.434 | 0.864 | 0.745 | 1.080 | 0.550 |
| 100. | 0.975 | 1.536 | 0.959 | 0.944 | 1.465 | 0.894 | 0.787 | 1.143 | 0.612 |
| 196. | 0.982 | 1.548 | 0.972 | 0.960 | 1.497 | 0.927 | 0.839 | 1.227 | 0.693 |
| 324. | 0.985 | 1.554 | 0.978 | 0.969 | 1.517 | 0.944 | 0.869 | 1.282 | 0.744 |
| 625. | 0.987 | 1.558 | 0.983 | 0.977 | 1.533 | 0.959 | 0.900 | 1.337 | 0.799 |
| 900. | 0.988 | 1.559 | 0.985 | 0.980 | 1.538 | 0.966 | 0.913 | 1.366 | 0.824 |

**Table II.2 - continued.**

| K | $V_0$ | | | | K | $V_0$ | | |
|---|---|---|---|---|---|---|---|---|
| | $\varphi = 0$ | $\varphi = 10$ | $\varphi = 10^3$ | $\varphi = 10^5$ | | $\varphi = 10^2$ | $\varphi = 10^4$ | $\varphi = 10^6$ |
| .144 | 0.0002 | 0.0002 | 0.0002 | 0.0001 | .16 | 0.002 | 0.0009 | 0.0001 |
| .196 | 0.018 | 0.017 | 0.012 | 0.003 | .25 | 0.051 | 0.021 | 0.003 |
| .256 | 0.065 | 0.061 | 0.041 | 0.010 | .36 | 0.133 | 0.059 | 0.010 |
| .4 | 0.195 | 0.183 | 0.122 | 0.033 | .64 | 0.297 | 0.144 | 0.029 |
| .625 | 0.348 | 0.327 | 0.227 | 0.068 | 1. | 0.428 | 0.228 | 0.055 |
| .9 | 0.471 | 0.443 | 0.317 | 0.111 | 1.96 | 0.604 | 0.379 | 0.116 |
| 1.6 | 0.638 | 0.604 | 0.459 | 0.200 | 3.24 | 0.708 | 0.491 | 0.162 |
| 3.6 | 0.804 | 0.772 | 0.640 | 0.346 | 6.25 | 0.810 | 0.625 | 0.251 |
| 6.4 | 0.879 | 0.852 | 0.742 | 0.447 | 9. | 0.853 | 0.690 | 0.306 |
| 10. | 0.918 | 0.896 | 0.806 | 0.528 | 16. | 0.904 | 0.768 | 0.396 |
| 19.6 | 0.956 | 0.940 | 0.876 | 0.642 | 36. | 0.948 | 0.854 | 0.520 |
| 32.4 | 0.973 | 0.960 | 0.913 | 0.715 | 64. | 0.967 | 0.897 | 0.603 |
| 62.5 | 0.986 | 0.977 | 0.945 | 0.793 | 100. | 0.977 | 0.922 | 0.659 |
| 90. | 0.990 | 0.983 | 0.957 | 0.828 | 196. | 0.986 | 0.948 | 0.732 |
| 160. | 0.994 | 0.989 | 0.971 | 0.872 | 324. | 0.990 | 0.962 | 0.777 |
| 360. | 0.997 | 0.995 | 0.983 | 0.916 | 625. | 0.994 | 0.975 | 0.826 |
| 640. | 0.999 | 0.997 | 0.989 | 0.936 | 900. | 0.996 | 0.980 | 0.849 |
| 1000. | 0.999 | 0.998 | 0.992 | 0.950 | | | | |

**Table II.3 - Numerically calculated values of $V_0$ as a function of K and $\phi$.**

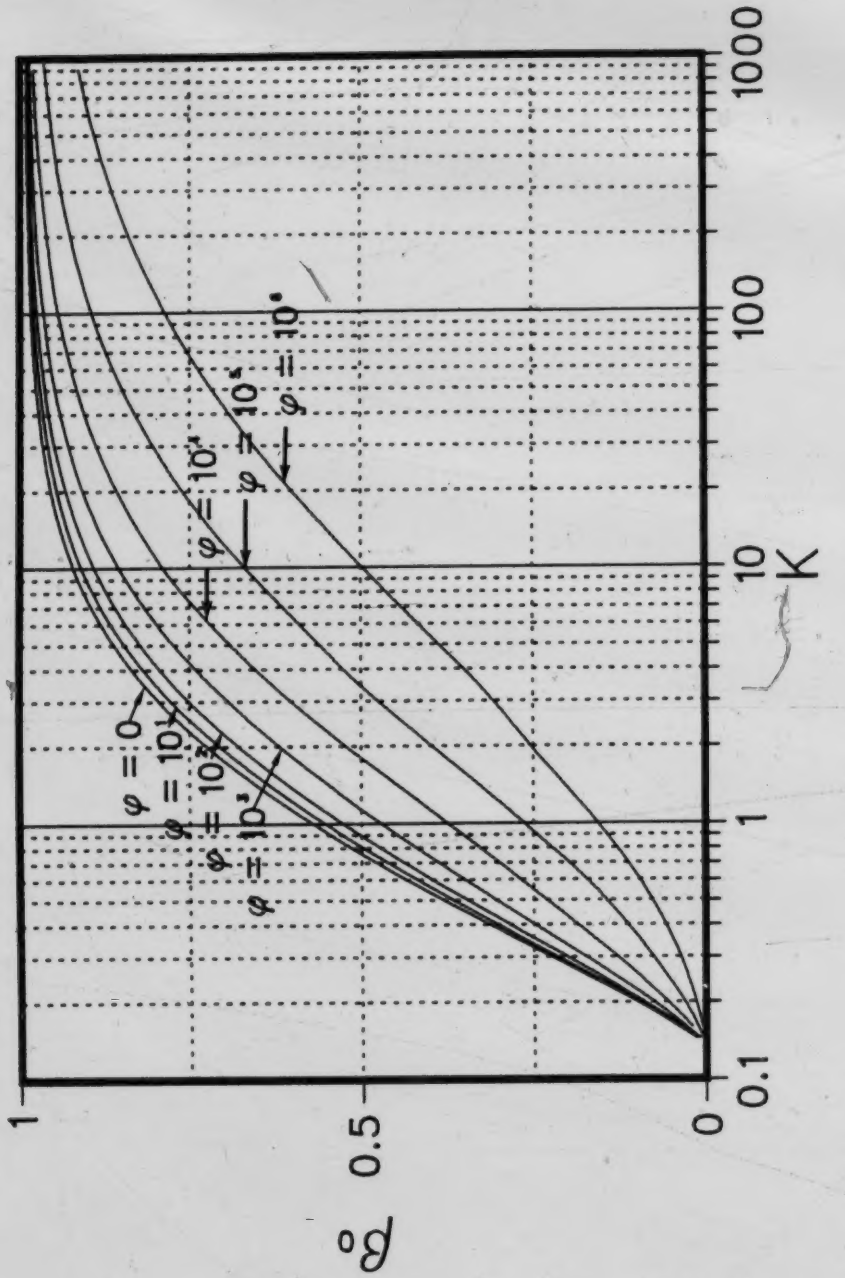



Figure II.3 - Smooth curves drawn for the numerical data of $\beta_0$ vs. K and $\phi$.

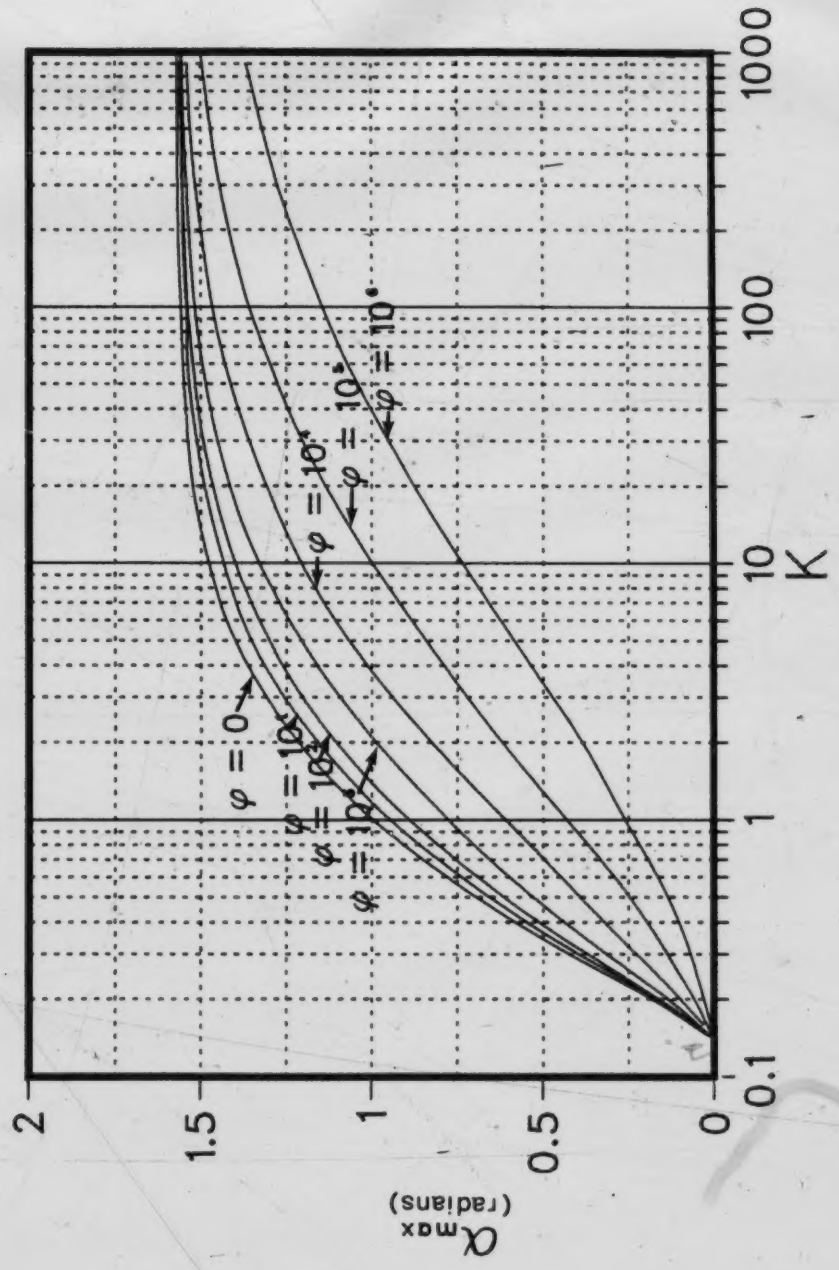



Figure II.4 - Smooth curves drawn for the numerical data of $\alpha_{max}$ vs. K and $\phi$.

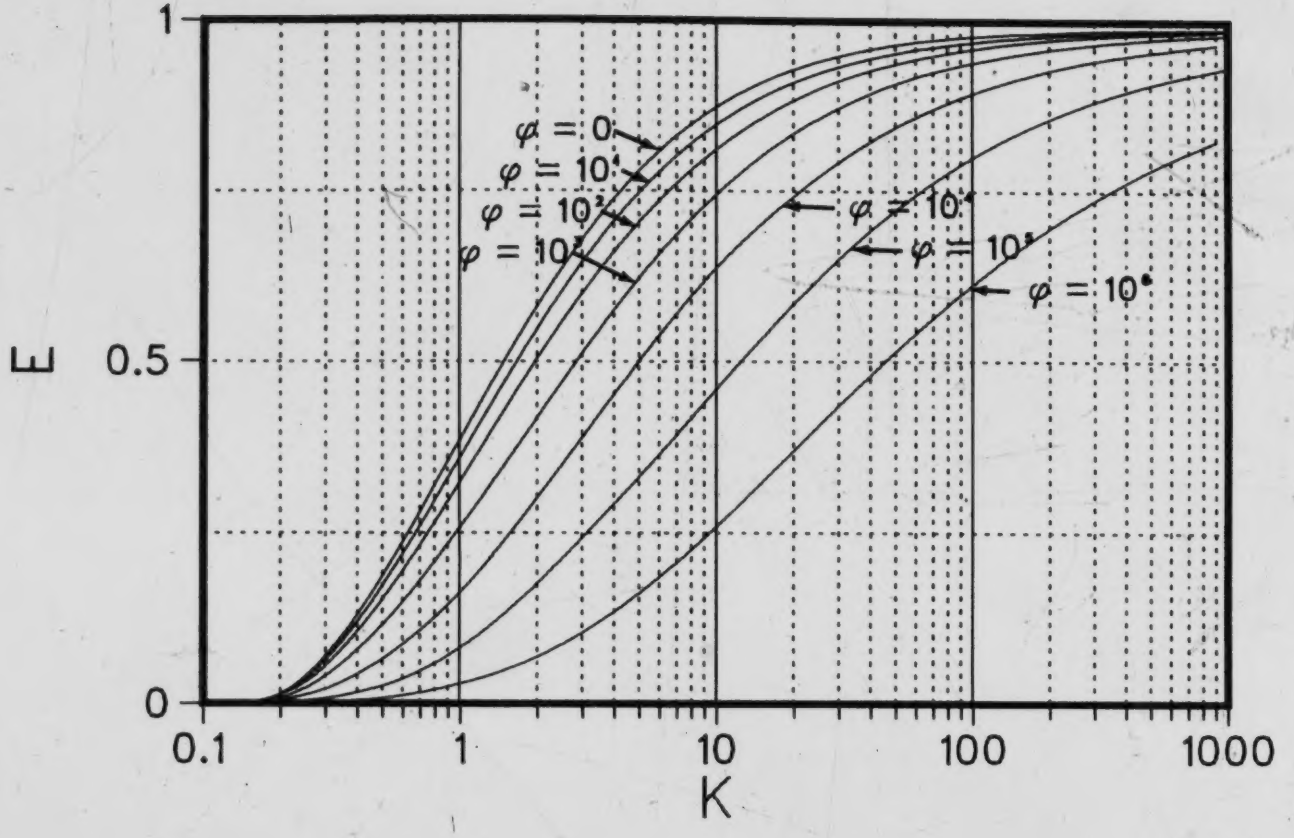



Figure II.5 - Smooth curves drawn for the numerical data of E vs. K and $\phi$.

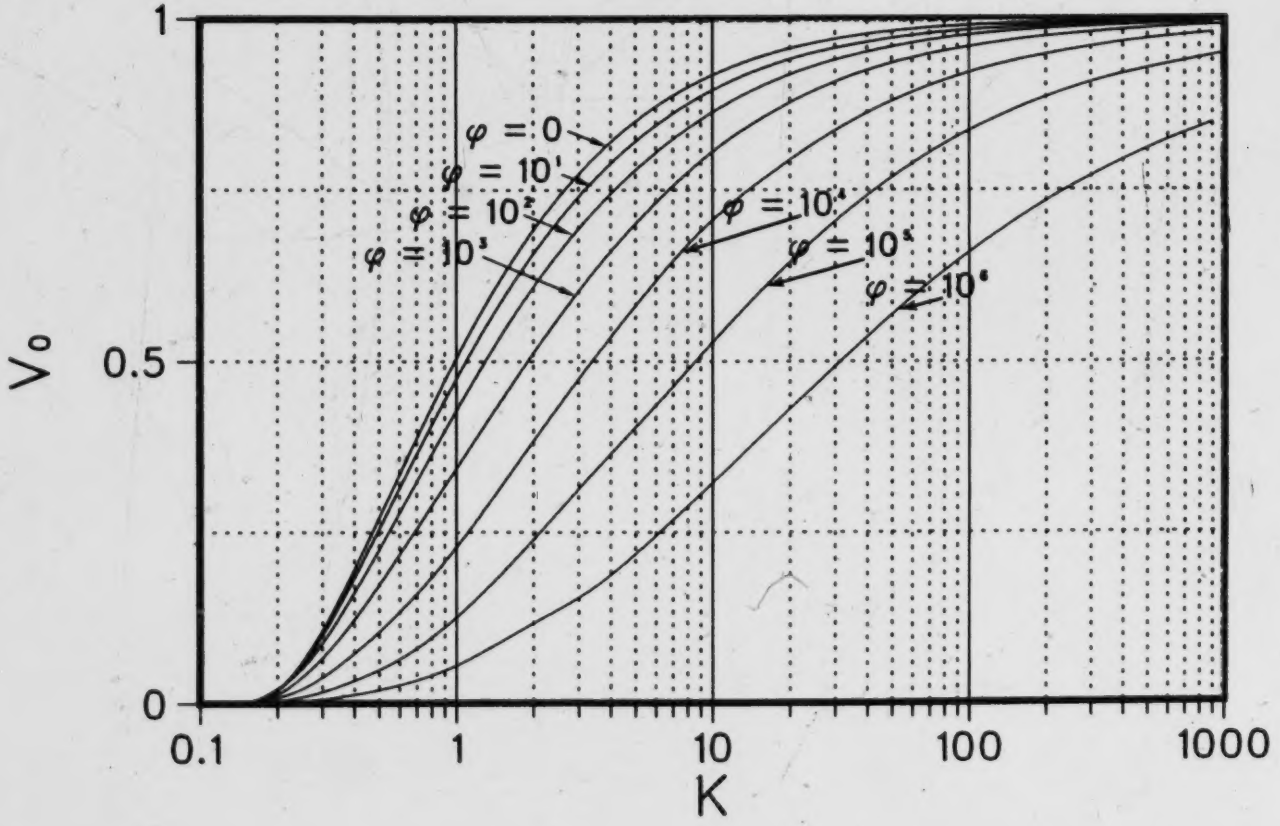



Figure II.6 - Smooth curves drawn for the numerical data of $V_0$ vs. K and $\phi$.

| Author(s) | $K_o(K,Re)$, $\zeta(U,D_d,D_c)$ | $E(K_o)$, $E(K,Re)$, $E(\delta)$ | Range |
|---|---|---|---|
| Langmuir and Blodgett | $K_o = .125 + \frac{(K - .125)}{1 + .0967\ Re^{.6367}}$ | $E = .466\ (\log_{10}(8K_o))^2$<br>$E = \frac{K_o}{K_o + \frac{\pi}{2}}$<br>$E = \frac{K}{K + \frac{\pi}{2}\ .112\ Re^{.63} + .75\ 10^{-4}\ Re^{1.38}}$ | $.125 < K_o < 1.1$<br>$K_o > 1.1$<br>$E > 0.5$ |
| Lozowski, Stallabrass and Hearty | $K_o = .125 + \frac{(K - .125)}{1 + .0967\ Re^{.6367}}$ | $E = .489\ (\log_{10}(8K_o))^{1.978}$<br>$E = \frac{K_o}{K_o + \frac{\pi}{2}}$ | $.125 < K_o < 0.9$<br>$K_o > 0.9$ |
| Cansdale and McNaughton | $K_0 = \frac{K}{1 + .087\ Re^{(.76\ Re^{-.027})}}$ | $E = .53\ (\log_{10}(8K_o))^{1.8}$<br>$E = \frac{K_o^{1.1}}{K_o^{1.1} + 1.223}$ | $.125 < K_o < 1.1$<br>$K_o > 1.1$ |
| Makkonen | $K_o = \frac{K}{1 + .087\ Re^{(.76\ Re^{-.027})}}$ | $E = .5\ (\log_{10}(8K_o))^{1.6}$<br>$E = \frac{K_o^{1.1}}{K_o^{1.1} + 1.426}$ | $.125 < K_o < 0.8$<br>$K_0 > 0.8$ |
| Stallabrass | $\zeta = \frac{U^{.6}\ D_d^{1.6}}{D_c}$ | $E = \frac{\zeta - 3200}{\zeta + 27000}$<br>$E = 0$ | $\zeta > 3200$<br>$\zeta < 3200$ |

**Table II.4 - Approximations to the numerical data of Langmuir and Blodgett derived by various authors.**

to various numerical icing models (Cansdale and McNaughtan, 1977, Stallabrass, 1980, Lozowski, Stallabrass and Hearty, 1983, Makkonen, 1984). A number of these fitted relations for $E(K,\phi)$ are listed in Table II.4.

In order to compare these relations, values computed from each are listed with a selection of the original data in Table II.5. The data all fall within a range of $\phi$ between $10^2$ and $10^4$, since some of the fitting functions were specifically designed for similar ranges, and because this range should apply to most physical applications, corresponding to to $10 \leq D_d \leq 4000$ microns, $0.01 \leq D_c \leq 1.0$ metre, $5 \leq U \leq 200$ metres per second and $-20°C \leq T \leq 0°C$. It is worth noting that the natural range is actually narrower than this, since the largest drops do not occur together with the highest wind speeds (e.g. marine icing), nor do the smallest droplets occur at the lowest speeds (e.g. aircraft icing).

Langmuir and Blodgett's own fit is the most accurate, since it includes a correction table for $E < 0.5$, and for larger values of K, (giving E values > 0.5), they give theoretically derived functions for $\beta_0$, $\alpha_{max}$, and E, which assume essentially straight-line trajectories. For this more elaborate scheme, the standard error (computed with respect to the numerical data) is generally an order of magnitude smaller than it is for the other fitting schemes. Even more important, however, is the fact that the variation among the values calculated by the various empirical

formulae for a given case can exceed 10 percent. These discrepancies will add to the difficulty of making comparisons among the icing predictions made by the various models which use these relations.

To reduce any further confusion, the new data have been fitted with approximating functions of improved form and greater accuracy. These functions are described below.

The approach taken by other authors to derive the formulae of Table II.4 has been to collapse a family of curves, such as that in Figure II.3, into a single curve via the ratio $K_0/K$, calculated as a function of Re, or in the case of Stallabrass (1980), via the parameter $\zeta$, a function of $D_d$, U, and $D_c$ (see Table II.4). $K_0$ is the K value read from the $\phi = 0$ curve which yields the same value of $\beta_0$, $\alpha_{max}$, E or $V_0$ as do the original K and $\phi$. Extensive manipulation of the new data has revealed major problems with this approach. Specifically, the three families of curves for $\beta$, $\alpha_{max}$ and E are not collapsible by the same ratio $K_0/K$, but require three different ratios. Only LB take this into account for small values of E, by using their correction table, and theoretical approximations for larger E. Furthermore, the $K_0/K$'s are *not* single-valued functions of Re (see Figure II.7), as has usually been assumed.

In the Stallabrass (1980) method, although $\zeta$ is used instead of K and $\phi$, a similar problem arises for his curves, since they collapse to a single curve only for very large values of $\zeta$.

| $\varphi$ | K | LB (numerical) | LB (approx.) | Lozowski, et al. | Cansdale | Makkonen | Stallabrass |
|---|---|---|---|---|---|---|---|
| $10^2$ | .16 | 0.004 | 0.004 | 0.004 | 0.002 | 0.005 | 0.056 |
| | .64 | 0.181 | 0.183 | 0.181 | 0.191 | 0.202 | 0.284 |
| | 3.24 | 0.591 | 0.588 | 0.568 | 0.630 | 0.594 | 0.634 |
| | 16. | 0.855 | 0.854 | 0.836 | 0.884 | 0.867 | 0.867 |
| | 100. | 0.966 | 0.965 | 0.958 | 0.976 | 0.972 | 0.966 |
| | 625. | 0.992 | 0.992 | 0.989 | 0.995 | 0.994 | 0.992 |
| $10^3$ | .196 | 0.009 | 0.009 | 0.010 | 0.0 | 0.0 | 0.017 |
| | .625 | 0.127 | 0.129 | 0.132 | 0.120 | 0.134 | 0.170 |
| | 3.6 | 0.542 | 0.540 | 0.509 | 0.560 | 0.522 | 0.536 |
| | 19.6 | 0.830 | 0.825 | 0.796 | 0.852 | 0.831 | 0.827 |
| | 90. | 0.942 | 0.939 | 0.925 | 0.956 | 0.949 | 0.943 |
| | 640. | 0.986 | 0.985 | 0.942 | 0.992 | 0.990 | 0.988 |
| $10^4$ | .25 | 0.012 | 0.013 | 0.014 | 0.0 | 0.0 | 0.0 |
| | 1. | 0.148 | 0.142 | 0.146 | 0.120 | 0.134 | 0.152 |
| | 6.25 | 0.546 | 0.539 | 0.499 | 0.557 | 0.518 | 0.531 |
| | 36. | 0.820 | 0.811 | 0.778 | 0.850 | 0.829 | 0.831 |
| | 196. | 0.937 | 0.934 | 0.921 | 0.960 | 0.954 | 0.951 |
| | 900. | 0.974 | 0.974 | 0.972 | 0.988 | 0.987 | 0.985 |
| | standard error: | | 0.003 | 0.018 | 0.017 | 0.010 | 0.019 |

**Table II.5 - Comparison of some original E values calculated numerically by Langmuir and Blodgett, and those derived from the approximations of Table II.4.**

For these reasons, an accurate fit to the entire range of data given is difficult to find and awkward to use. Consequently, an approximating function is given here only for $\phi$ between $10^2$ and $10^4$, which is the range applicable to most atmospheric, land-based and marine icing problems.

A non-linear regression analysis on this range gives the following result for $0.17 \leq K \leq 10^3$:

$$X(K,\phi) = (C_{X,1} K^{C_{X,2}} \exp(C_{X,3} K^{C_{X,4}} + C_{X,5}) - (C_{X,6}(\phi - 100)^{C_{X,7}}) (C_{X,8} K^{C_{X,9}} \exp(C_{X,10} K^{C_{X,11}}) + C_{X,12}), \quad \text{II.11}$$

where X is either $\beta_0$, $\alpha_{max}$, E or $V_0$, and the constants $C_{X,n}$ are as listed in Table II.6. The basic form of this function is a standard fitting function supplied by the BMDP non-linear regression package (Dixon, et al., 1981).

Values for $K < 0.17$ and $K > 10^3$ may be read from the curves or *roughly* approximated as follows:

$$\beta_0 = \alpha_{max} = E = V_0 = 0.0, \text{ for } K < 0.17, \quad \text{II.12}$$

and

$$\beta_0 = E = V_0 = 0.99, \text{ for } K > 10^3,$$

$$\alpha_{max} = 1.56, \text{ for } K > 10^3. \quad \text{II.13}$$

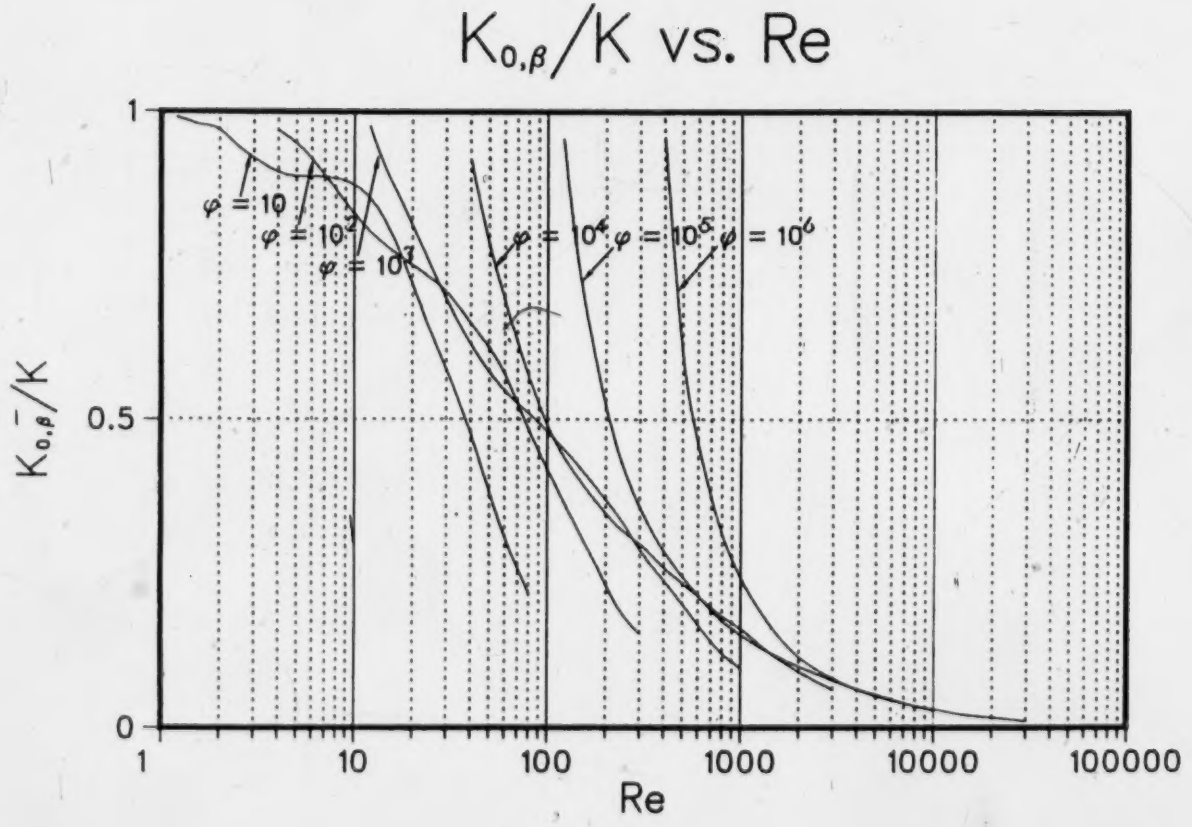



Figure II.7 The dependence of $\frac{K_{0,\beta}}{K}$ on Reynolds Number Re. $K_{0,\beta}$ is defined by: $\beta_0(K,\varphi) = \beta_0(K_{0,\beta}, 0)$

| | $X = \beta_o$ | $X = \alpha_{max}$ | $X = E$ | $X = V_o$ |
|---|---|---|---|---|
| $C_{X,1}$ | 1.218 | 2.433 | 1.066 | 1.030 |
| $C_{X,2}$ | $-6.70*10^{-3}$ | $-4.70*10^{-3}$ | $-6.16*10^{-3}$ | $1.68*10^{-3}$ |
| $C_{X,3}$ | -0.551 | -0.375 | -1.103 | -0.796 |
| $C_{X,4}$ | -0.643 | -0.576 | -0.688 | -0.780 |
| $C_{X,5}$ | -0.170 | -0.781 | -0.028 | -0.040 |
| $C_{X,6}$ | $3.05*10^{-3}$ | $8.50*10^{-3}$ | $6.37*10^{-3}$ | $9.44*10^{-3}$ |
| $C_{X,7}$ | 0.430 | 0.383 | 0.381 | 0.344 |
| $C_{X,8}$ | 2.220 | 1.757 | 3.641 | 2.657 |
| $C_{X,9}$ | -0.450 | -0.298 | -0.498 | -0.519 |
| $C_{X,10}$ | -0.767 | -0.420 | -1.497 | -1.060 |
| $C_{X,11}$ | -0.806 | -0.960 | -0.694 | -0.842 |
| $C_{X,12}$ | -0.068 | -0.179 | -0.045 | -0.029 |

Table II.6 Coefficients for approximating impingement parameters by the expression:

$$X(K,\varphi) = \left(C_{X,1}\, K^{C_{X,2}} \exp(C_{X,3}\, K^{C_{X,4}}) + C_{X,5}\right) \cdot \left(C_{X,6}(\varphi - 100)^{C_{X,7}}\right)\left(C_{X,8}\, K^{C_{X,9}} \exp(C_{X,10}\, K^{C_{X,11}}) + C_{X,12}\right)$$

Within the range of $0.17 < K < 10^3$ the standard error of the fit with respect to the numerical data is given in Table II.7. It compares very favorably to the errors given in Table II.5 for the fitting schemes previously discussed.

**E. Limitations of LB Theory**

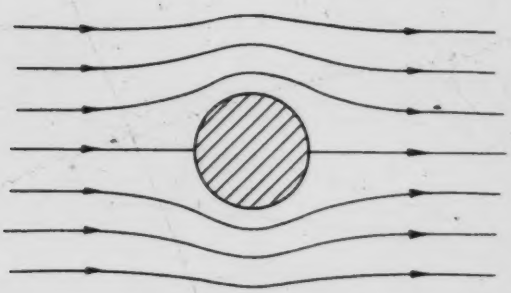
The collision efficiency theory of Langmuir and Blodgett assumes that the air is in potential flow about the collecting cylinder, that only steady-state drag forces are exerted on the water droplets by the flow, and that the flow itself remains unaffected by the presence of the droplets.

The potential flow field about an infinitely long, circular cylinder is as described by Equations II.7 , and illustrated in Figure II.8(a). Figure II.8(b) illustrates the analogous situation for viscous flow at a cylinder Reynolds number of $\simeq 10^4$, corresponding to $D_c = 0.0254$ m, U = 10 m/sec, and T = -10° C. The upstream flow in Figure II.8(b), which determines the accretion, deviates slightly from the potential flow case due to the presence of a sizable wake on the downstream side. The cylinder plus wake will be felt by the flow further upstream, decreasing the streamline curvature, and thus increasing the collision efficiency. In most situations, however, the effect will be small.

The potential flow assumption also ignores the cylinder boundary layer, which will be of negligible depth for a

| | | numerical | | | | approximated | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\varphi$ | K | $\beta_0$ | $\alpha_{max}$ | E | $V_0$ | $\beta_0$ | $\alpha_{max}$ | E | $V_0$ |
| $10^2$ | .16 | 0.034 | 0.054 | 0.001 | 0.002 | 0.036 | 0.054 | 0.0 | 0.0 |
| | .64 | 0.415 | 0.720 | 0.212 | 0.297 | 0.416 | 0.720 | 0.210 | 0.293 |
| | 3.24 | 0.763 | 1.219 | 0.617 | 0.708 | 0.763 | 1.219 | 0.619 | 0.710 |
| | 16. | 0.921 | 1.446 | 0.863 | 0.904 | 0.920 | 1.445 | 0.861 | 0.904 |
| | 100. | 0.975 | 1.536 | 0.959 | 0.977 | 0.978 | 1.538 | 0.961 | 0.975 |
| | 625. | 0.987 | 1.558 | 0.983 | 0.994 | 0.986 | 1.558 | 0.983 | 0.995 |
| $10^3$ | .196 | 0.073 | 0.131 | 0.007 | 0.012 | 0.075 | 0.135 | 0.005 | 0.013 |
| | .625 | 0.359 | 0.612 | 0.157 | 0.227 | 0.363 | 0.611 | 0.159 | 0.220 |
| | 3.6 | 0.725 | 1.136 | 0.553 | 0.640 | 0.728 | 1.139 | 0.557 | 0.639 |
| | 19.6 | 0.906 | 1.413 | 0.832 | 0.876 | 0.904 | 1.401 | 0.825 | 0.869 |
| | 90. | 0.963 | 1.512 | 0.935 | 0.957 | 0.964 | 1.503 | 0.931 | 0.951 |
| | 640. | 0.984 | 1.553 | 0.976 | 0.989 | 0.983 | 1.549 | 0.974 | 0.989 |
| $10^4$ | .25 | 0.094 | 0.168 | 0.012 | 0.015 | 0.098 | 0.179 | 0.012 | 0.024 |
| | 1. | 0.374 | 0.609 | 0.163 | 0.228 | 0.379 | 0.610 | 0.162 | 0.225 |
| | 6.25 | 0.726 | 1.115 | 0.547 | 0.625 | 0.726 | 1.115 | 0.548 | 0.637 |
| | 36. | 0.899 | 1.379 | 0.812 | 0.854 | 0.899 | 1.380 | 0.816 | 0.865 |
| | 196. | 0.960 | 1.497 | 0.927 | 0.948 | 0.962 | 1.497 | 0.930 | 0.954 |
| | 900. | 0.980 | 1.538 | 0.966 | 0.980 | 0.980 | 1.543 | 0.967 | 0.986 |
| | standard error: | | | | | 0.002 | 0.005 | 0.003 | 0.006 |

Table II.7 - Comparisons between a selection of numerically calculated values, and approximated values calculated according to Table II.6.



(a)

(b)

Figure II.8 - (a):Potential flow streamlines about a cylinder in cross-section. (b):Viscous flow streamlines about a cylinder at a Reynolds number of 10,000.

smooth cylinder, but may become significant for rough surfaces. The effect, if any, of a deep, possibly turbulent boundary layer on the collision efficiency is unknown. In particular, the onset of accretion may be affected if the roughness elements are large enough to act as individual collectors.

The assumption of the airflow being undisturbed by the droplets carried in it, is a reasonable one given typical cloud liquid water contents.

In determining the deceleration of the drop by the flow, LB's steady-state viscous drag coefficient ignores buoyancy and gravitational forces, the influence of vorticity diffusion from the accelerating droplet surface, and the induced mass increase of the drop as a result of accelerating the air immediately surrounding it.

A 20 micron droplet falls in air with a terminal speed of 0.012 m/sec (Mason, 1971). Thus the vertical deviation of such a droplet approaching a 0.0254 m diameter cylinder from five diameters upstream, at a free-stream speed of 10 m/sec, is about 0.15 mm. This is only about one percent of the maximum vertical deviation due to the flow, under those conditions. At some point on the cylinder surface, the deviation due to the flow will be comparable to the gravitational drift, but since the relative difference in two adjacent trajectories will be much smaller, the collision efficiencies should not be affected.

Much larger drops, such as occur in sea spray icing, will have a correspondingly larger gravitational drift. Preliminary numerical calculations by Szilder (personal communication, 1986) indicate that there is still a very small effect on the collision efficiency, but this is a question which should be investigated further.

The other two effects on the drag are incorporated into the 'history' term, which, expressed as an acceleration, is:

$$-\frac{18\rho_a}{(2\rho_d+\rho_a)D_d}\sqrt{\frac{\mu}{\pi\rho_a}}\int_{-\infty}^{t}\frac{d\vec{v}}{d\tau}\frac{d\tau}{\sqrt{t-\tau}} \qquad \text{II.14}$$

The relative importance of the history term is often measured by the acceleration modulus (Norment, 1980):

$$N_A = D_d \left|\frac{dv}{dt}\right| / v^2, \qquad \text{II.15}$$

where v is the droplet's absolute velocity. If this quantity is < 0.01, the history term may be safely omitted from the droplet's equation of motion.

Numerical results from the trajectory integration model presented above show that $N_A$ is well within this limit under most conditions of accretion on cylinders. However, for $K \leq 0.5$, $N_A$ may reach values $\geq 0.01$ just before impact, and for $K \leq 0.20$, the limit is exceeded as much as a few millimetres in front of the cylinder surface.

For these small K values then, the effect of the history term is to decrease the droplet's deceleration and

increase both its total velocity near the surface, and the resulting collision efficiency. Numerical integrations by Oleskiw (1982) for K = 0.196, $\phi$ = 1000 show an increase in E of 350%, and in $\beta_0$ of 167% when the history term is included.

Thus the 'cut-off' K value, for zero collision efficiency, which arises from LB theory, is not valid for real flow. In light of these results, it may be advisable to consider Equations II.11, II.12, and II.13 as valid only for K > 0.25. Below this limit, the trajectories should be recalculated using the complete drag terms in the droplet equation of motion.

The last assumption listed, that of an airflow undisturbed by the presence of water droplets, is probably a reasonable one for th small liquid water contents typical of clouds, < 5 $g/m^3$. The case for heavy spray icing may, however, be very different, and this is another question which warrants further study.

**F. Summary**

Earlier results for water droplet trajectories at freezing temperatures have been re-examined and improved upon. The new results differ from LB by as much as 10 percent. Tabulated data are provided to replace those of Langmuir and Blodgett for the collisional parameters of water droplets on cylinders, and a good approximating function to the data has been found which will be of use to

modellers. These results should only be applied, however, to real situations when $K > 0.25$, below which the non-steady-state drag terms have a significant effect on collision efficiencies.

## References


Beard, K.V. 1976: Terminal Velocity and Shape of Cloud and Precipitation Drops Aloft. *J. Atmos. Sci.* , 33, 851-864. *J. Atmos. Sci.*,33, 851-864.

Beard, K.V. and Pruppacher, H.R. 1969: A Determination of the Terminal Velocity and Drag of Small Water Droplets by Means of a Wind Tunnel. *J. Atmos. Sci.*, 26, 1066-1072.

Cansdale, J.T. and McNaughton, I.I. 1977: Calculation of Surface Temperature and Ice Accretion Rate in a Mixed Water Droplet / Ice Crystal Cloud. Royal Aircraft Establishment Technical Report 77090, 29 pp.

Dixon, W.J., Chief Editor, et. al. 1980: Chapter 14 - Nonlinear Regression. In: BMDP Statistical Software. University of California Press, 289-309.

Green, S. L. 1937: Hydro-and-Aero-Dynamics. Pitman and Sons, p. 26.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories.

Collected Works of Irving Langmuir Vol. 10, Pergamon Press, 348-393.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Non-Rotating Cylinder Part I: A Simulation Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

Mesinger, F. and Arakawa, A. 1976: Numerical Methods used in Atmospheric Models, Vol. 1. W.M.O. GARP Publications Series No. 17, 64 pp.

Mason, B.J. 1971: The Physics of Clouds, 2nd Edition. Clarendon Press, p. 594.

Makkonen, L. 1984: Modelling of Ice Accretion on Wires. *J. Climate and App. Meteor.*, 23, 929-939.

Norment, H.G. 1980: Calculation of Water Drop Trajectories to and About Three-Dimensional Bodies in Potential Airflow. NASA Contractor Report 3291, 82 pp.

Oleskiw, M.M. 1982: A Computer Simulation of Time-Dependent Rime Icing on Airfoils. Ph.D. thesis, The University of Alberta, 302 pp.

Stallabrass, J.R. 1980: Trawler Icing: A Compilation of Work

Done at NRC. NRC Report MD-56, NRC No. 19372, 103 pp.

# III. An Operational Model for Rime Ice Accretion

## A. Introduction

Modelling of the many physical processes involved in ice accretion can be a complex and time-consuming exercise. The cost of running such models tends to limit their use as investigative tools, or for predictive purposes of an operational character. In an attempt to alleviate part of this problem, simple parameterization methods have been developed which make possible a more practical, inexpensive and realistic model. In this chapter these methods, and their application in a computer simulation of rime icing on circular cylinders will be described.

The computer model was developed with several specific purposes in mind. The first was to reduce the computational time necessary, in particular, for the calculation of impingement parameters. The quantities E, $\beta_0$ and $\alpha_{max}$ for circular cylinders are available from the tabulated data and analytical fitting functions of Chapter II. Values of local collision efficiency between the stagnation line and maximum impingement angle usually have to be determined from a series of droplet trajectory pairs.[1] Oleskiw (1982) has calculated the distributions of local collision efficiency

---

[1] It has been suggested (McComber and Touzot, 1981) that the variation of the local radial velocity of impact be used as an approximation to the local collision efficiency. It has been previously shown (Lozowski, Finstad and Gates, 1985) that for realistic cases this can be a poor approximation at the stagnation line (compare Figures II.3 and II.6), and that the relative error increases dramatically with distance toward the accretion limit.

for several cases, from numerically integrated trajectories. From these, it was possible to derive a parameterized general form of the $\beta(\alpha)$ curve which depends only on the parameters $\beta_0$ and $\alpha_{max}$.

Use of this parameterization is the major time-saving device in the model. Several other options are provided for simulating the effect on the $\beta(\alpha)$ distribution of a droplet size spectrum which cannot be adequately modelled by the median volume size. This effect is principally in the extension of the maximum accretion angle.

A second major purpose in developing this model was to incorporate the local ice density variations into the shape of the rime ice accretions. An empirical, parameterized function is used to describe this variation, and it is shown to be reasonably successful in reproducing observed accretion shapes on non-rotating cylinders under a variety of conditions.

A third purpose was to attempt to simulate a slow rotation of the collecting cylinder about its long axis. This is meant to imitate the effects of a gravitational torque on an ice-loaded cable. Although this capability has now been built into the model, it has not yet been been possible to test it against laboratory or field data.

In order to effectively examine the validity of these methods, only dry, rime ice accretion is considered. This is so that any comparisons with experimental results are not dependent on poorly known quantities, such as heat transfer,

which affect wet icing processes.

In the experimental accretion experiments used to aid in model development and verification, a crucial measurement is the droplet size spectrum, and its median volume diameter. All droplet spectra were measured by the oil slide method, an outline of which is given in Mason (1971). However, the mvd's derived from such measurements have been found to show a systematic error by Makkonen and Stallabrass (1984). We therefore apply their empirical correction formula to our data in all but a few cases, which fall outside the range of validity for the correction. The procedure followed in these cases is described in Appendix III.

To ensure rime icing under the conditions typical of the wind-tunnel experiments, in which collectors with diameters of 2 to 5 cm were exposed to typical atmospheric liquid water contents and wind speeds, the air temperature in the model is assumed to be always at or below -10° C. For application to general conditions, the model should include a calculation of the critical liquid water content, or 'Ludlam limit', (Ludlam, 1951) in order to determine whether the ice will be wet or dry.

**B. Parameterization for Local Collision Efficiency**

For the idealized situation of straight-line droplet trajectories the local collision efficiency is given simply by cos $\alpha$, where $\alpha$ is the angle of the local surface normal

with respect to the free stream direction. The collision efficiency has a maximum value of unity at the stagnation line, and a minimum of zero at $\alpha = \pi/2$.

In a more realistic situation, the trajectories curve away from the free stream direction as they approach the cylinder. The stagnation line collision efficiency is reduced from 1, and the local value reaches zero for some $\alpha = \alpha_{max} < \pi/2$. This situation can be well approximated (especially for large droplets whose trajectory curvature is small) by directly scaling the cos $\alpha$ distribution to these two extreme values, as follows:

$$\beta(\alpha) = \beta_0 \cos ((\alpha/\alpha_{max}) (\pi/2)), \qquad \text{III.1}$$

where the two parameters $\beta_0$ and $\alpha_{max}$ are determined from tabulated values or approximating functions given in Chapter II.

A form of time dependence is built into this method since the collision efficiency at a point depends on its local surface angle, which changes with time as the ice layers are accreted on top of one another. The parameters $\beta_0$ and $\alpha_{max}$ do not change with time, an approximation which becomes worse as the accretion shape grows, and departs from a circular cross-section.

For smaller droplets the simple approximation of Equation III.1 fails, since the trajectories are curved by much larger amounts. It has been found that, for a $\beta_0$ <

0.65, the $\beta(\alpha)$ curve is more closely approximated by:

$$\beta(\alpha) = \beta_0 - (\beta_0 (1 - \cos^{0.5}\alpha)/(1 - \cos^{0.5}\alpha_{max})). \quad \text{III.2}$$

For a 0.0254 m diameter cylinder at typical ground level wind speeds of a few tens of metres per second, this corresponds to water droplets of less than 20 microns diameter.

Figure III.1 illustrates the goodness of fit of these functions against local collision efficiency curves calculated numerically by Oleskiw (1982), and approximated by Lozowski, Stallabrass and Hearty (1983).

In both Equations III.1 and III.2, a monodisperse droplet size population has been assumed, or at least a distribution for which the collision efficiency may be reasonably approximated by the collision efficiency of the median volume droplet, or mvd. The results of Appendix VI suggest that, for typical ground-level wind speeds, the single droplet size approximation holds well for $\beta_0 \geq 0.5$.

Otherwise, the effect of the size spectrum, and particularly its larger drops, should be accounted for. Droplet size distributions for icing purposes are commonly described by volume percentages in several size bins of finite width (see, for example, Figure III.2). Applying either Equation III.1 or III.2 to the median volume droplet of *each size bin*, a volume-weighted average distribution may be calculated. Such a weighted curve then illustrates the

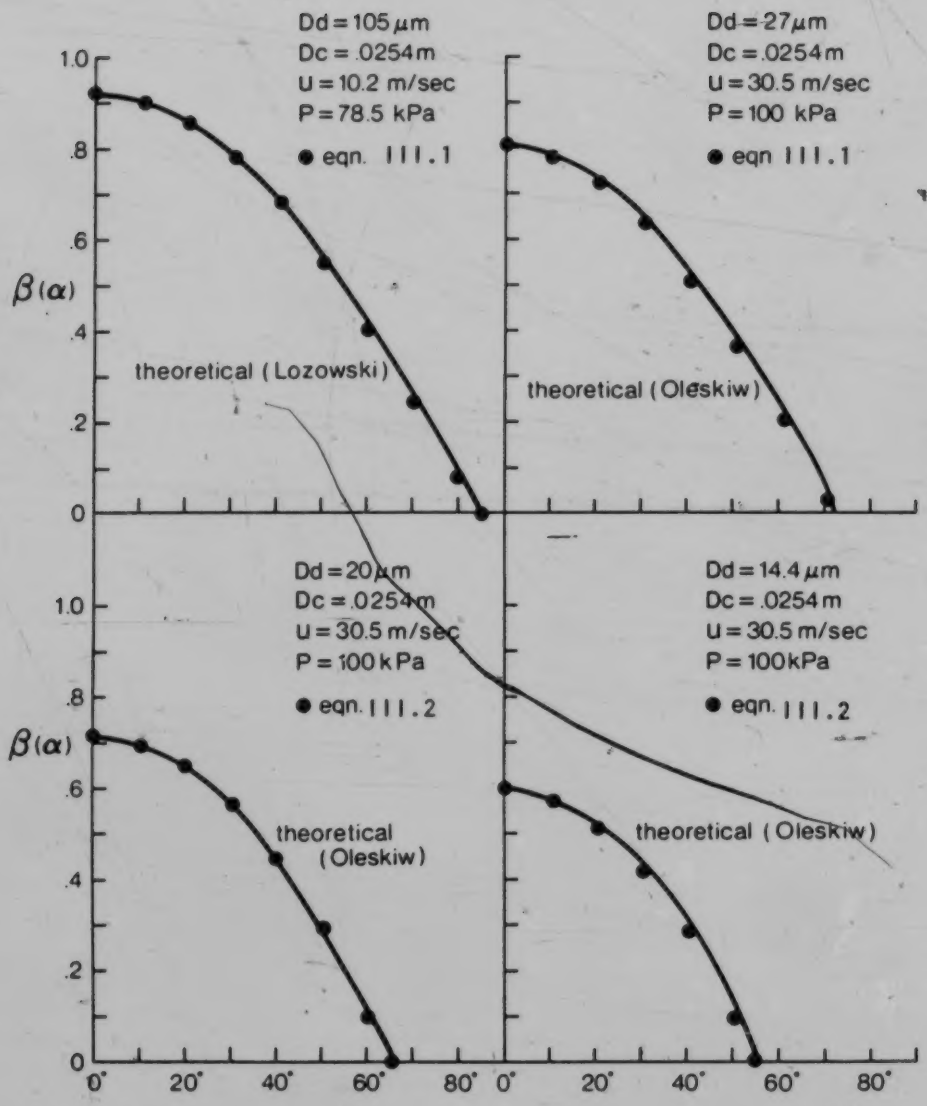



Figure III.1 - Parameterized $\beta(\alpha)$ values from Equations III.1 and III.2, compared to numerically calculated curves of Oleskiw (1982), and parameterized curve after Lozowski, Stallabrass, and Hearty (1983).

effect of the smaller droplets in extending the maximum accretion limit.

This weighting procedure, for a given input size spectrum, may be carried out within the model. This is the "weighted spectrum" option. However, the nature of the averaging procedure eventually gives rise to first-order discontinuities in the $\beta(\alpha)$ curve, and thus on the accretion surface. For this reason, the weighted spectrum option is not recommended when more than three ice layers are accreted, no matter how thin these layers are.

An alternative option is included to avoid this problem, which has been called the "simulated spectrum" option. It is another form of the $\beta(\alpha)$ curve, this time based on weighted average curves (from laboratory measured spectra) which have been scaled and fitted with polynomial functions.

for $\beta_0 < 0.65$:

$$\beta(\alpha/\alpha_m) = \beta_0 \, (1 + 0.22(\alpha/\alpha_m) - 3.08(\alpha/\alpha_m)^2 + 1.86(\alpha/\alpha_m)^3), \qquad \text{III.3}$$

and for $\beta_0 > 0.65$:

$$\beta(\alpha/\alpha_m) = \beta_0 \, (1 + 0.029(\alpha/\alpha_m) - 1.94(\alpha/\alpha_m)^2 + 2.48(\alpha/\alpha_m)^3 - 4.11(\alpha/\alpha_m)^4 + 2.54(\alpha/\alpha_m)^5), \qquad \text{III.4}$$

where $\alpha_m = \alpha_{max} + Tl$, and Tl is the "tail length", an extension to $\alpha_{max}$, derived from numerical results and correlated to the mvd. For $\beta_0 < 0.65$:

$$Tl = 0.52 \exp(-mvd / 70) \qquad III.5$$

and for $\beta_0 > 0.65$:

$$Tl = 0.37 \exp(-mvd / 70). \qquad III.6$$

Here mvd is in $\mu m$ and Tl is in radians. Equations III.3 and III.4 are compared in Figure III.2 with two weighted average curves for the cloud droplet spectra shown.

Although these methods may seem to have become very superficial, they do retain the desired economy of computation with an acceptably small loss of accuracy.

### C. Variation of Local Density

A study of the local density variations in circular cylinder rime accretions will be discussed in some detail in Chapter IV. Here the result is presented briefly; an empirical curve describing local density as a function of surface slope.

Based on cross-sectional thickness measurements of several shallow rime layers grown in the laboratory under known conditions, the following function was derived for $\alpha/\alpha_m < 1$:

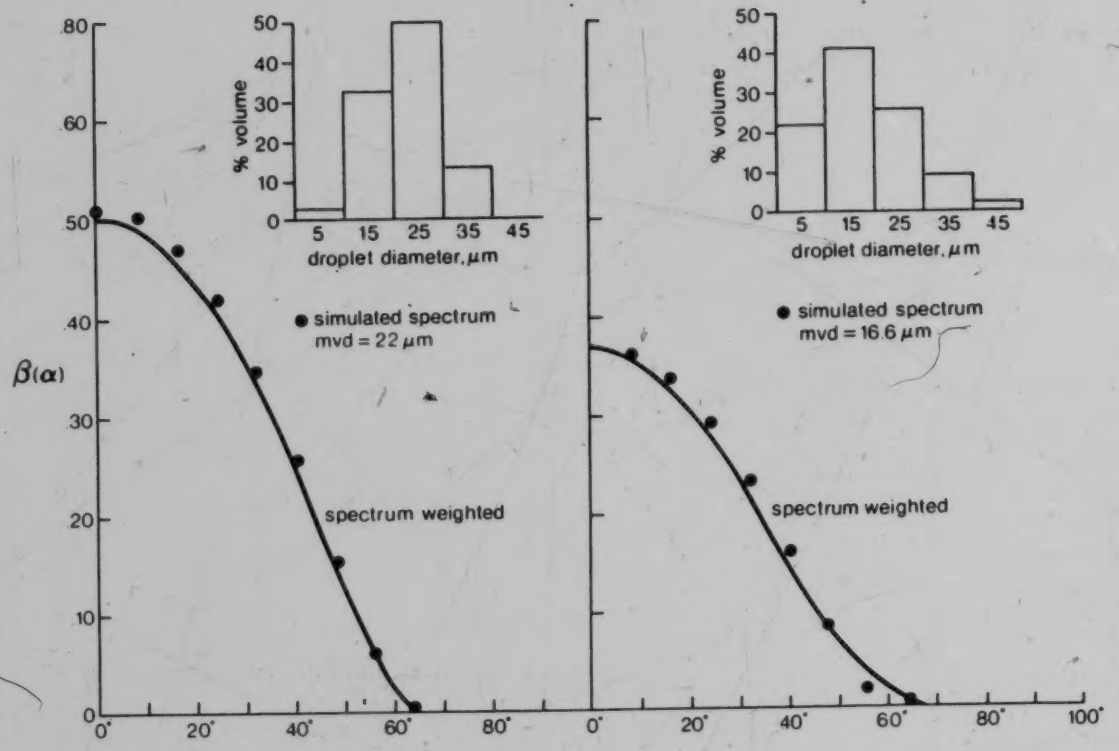



Figure III.2 - Parameterized $\beta(\alpha)$ values from Equations III.3 and III.4, compared to weighted average curves for the cloud droplet spectra shown. These spectra are derived from the distribution of Best (1951).

$$\rho/\rho_0 = (1 - 0.143(\alpha/\alpha_m) - 0.246(\alpha/\alpha_m)^2 - 0.309(\alpha/\alpha_m)^3). \qquad \text{III.7}$$

Here $\alpha_m$ is defined as for Equations III.3 and III.4, and in this version of the model, $\rho_0$ = 890 kg / $m^3$. This assumption is made for simplicity; in reality the rime densities may be much lower, and this will affect the accretion thickness and overall shape. Chapter IV will discuss methods for estimating more realistic values of $\rho_0$.

The remainder of this section will examine the predictive success of the model under a variety of input conditions. Figure III.3 compares accretion profiles and masses from the model against ice accretions grown in wind tunnel experiments carried out in the University of Alberta's FROST tunnel (this facility and our experimental methods are described in Gates, 1981 and Lozowski and Gates, 1984). For the ranges shown of mvd, liquid water content and free-stream speed the modelled mass is always within 10% of the observed mass, and the major features of the experimental shapes are reproduced in the profiles.

However, it is not expected that the model will reproduce such features as ridges or troughs near the stagnation line. These are most likely due to changes in the flow pattern as the accretion grows, and the resulting

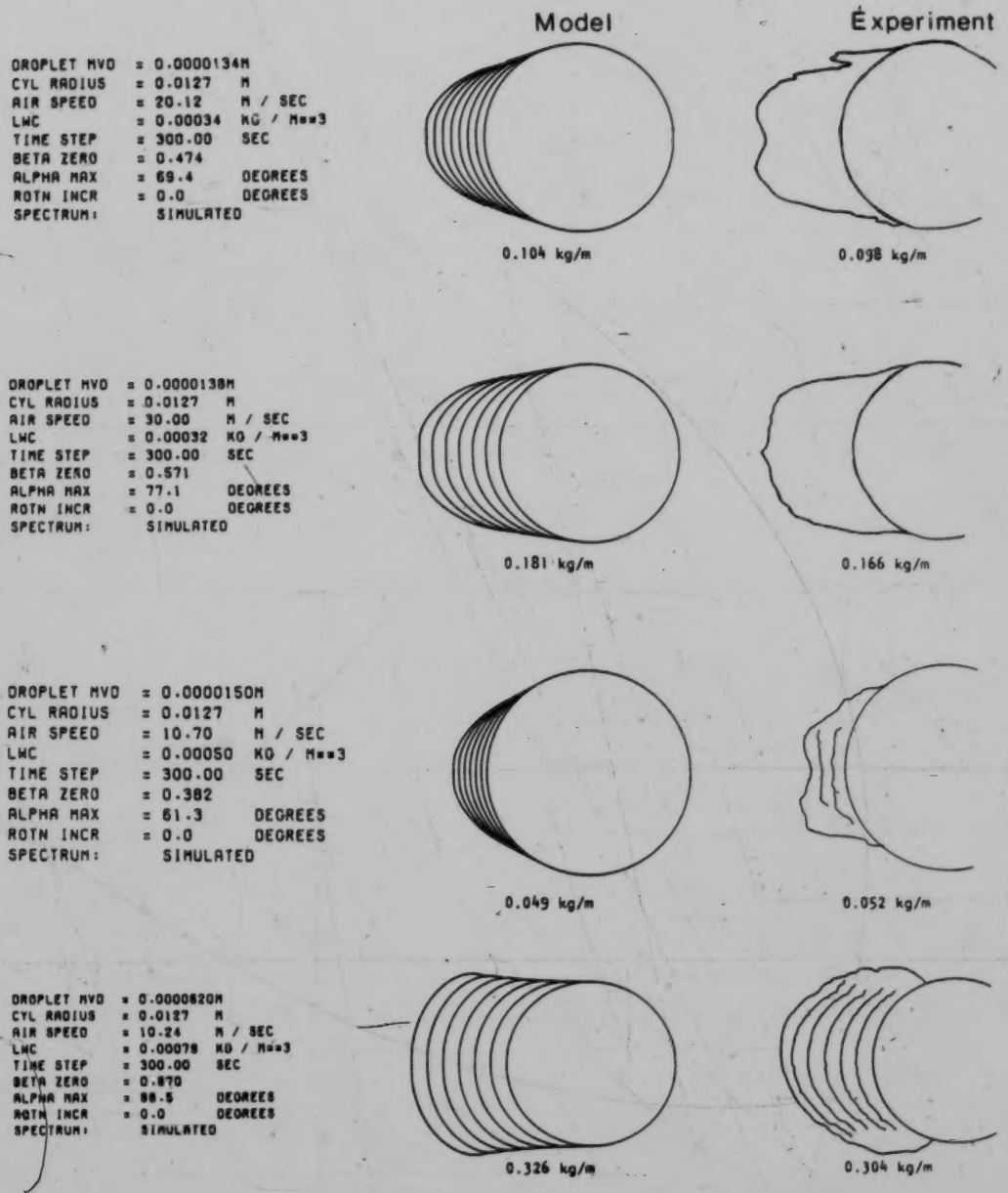

Model
Experiment
DROPLET MVD = 0.0000134M
CYL RADIUS = 0.0127 M
AIR SPEED = 20.12 M / SEC
LWC = 0.00034 KG / M**3
TIME STEP = 300.00 SEC
BETA ZERO = 0.474
ALPHA MAX = 69.4 DEGREES
ROTN INCR = 0.0 DEGREES
SPECTRUM: SIMULATED
0.104 kg/m
0.098 kg/m
DROPLET MVD = 0.0000138M
CYL RADIUS = 0.0127 M
AIR SPEED = 30.00 M / SEC
LWC = 0.00032 KG / M**3
TIME STEP = 300.00 SEC
BETA ZERO = 0.571
ALPHA MAX = 77.1 DEGREES
ROTN INCR = 0.0 DEGREES
SPECTRUM: SIMULATED
0.181 kg/m
0.166 kg/m
DROPLET MVD = 0.0000150M
CYL RADIUS = 0.0127 M
AIR SPEED = 10.70 M / SEC
LWC = 0.00050 KG / M**3
TIME STEP = 300.00 SEC
BETA ZERO = 0.382
ALPHA MAX = 61.3 DEGREES
ROTN INCR = 0.0 DEGREES
SPECTRUM: SIMULATED
0.049 kg/m
0.052 kg/m
DROPLET MVD = 0.0000820M
CYL RADIUS = 0.0127 M
AIR SPEED = 10.24 M / SEC
LWC = 0.00078 KG / M**3
TIME STEP = 300.00 SEC
BETA ZERO = 0.870
ALPHA MAX = 88.5 DEGREES
ROTN INCR = 0.0 DEGREES
SPECTRUM: SIMULATED
0.326 kg/m
0.304 kg/m

Figure III.3 - Comparisons between modelled and experimental profiles and masses per unit length for the conditions listed at left. The lower two experimental profiles have layer outlines which were marked by spray paint.

changes in the impingement characteristics.

In particular, the value of $\beta_0$ will change with time, while the model assumes a constant value. Even with this assumption, however, it is able to distinguish between widening and narrowing profiles for small differences in growth conditions, and can produce the flat front surface which often precedes the formation of ridges and troughs in the experiments.

**D. Rotation**

An option has been included in the model which will allow the cylinder plus accretion to be rotated with respect to the free stream by a small amount after each layer is accreted. No attempt is made to calculate the correct rotation rate due to the gravitational and aerodynamic torque; the rotation speed in degrees per layer must be specified as one of the input parameters.

In order to achieve this effect within the model algorithm, the local collision efficiency and local density must be calculated as functions of the position angle rather than the surface angle on the current surface. (The position angle at a surface point is the angle between a line joining the point to the centre of the cylinder, and the free-stream direction.) This somewhat less accurate method was adopted in order to avoid problems when accreting over top of the previous layers' accretion limit.

There are presently no laboratory or field data available with which to test this part of the model. It is presented here only as an option for future study. An example of a rotating case is shown in Figure III.4.

The Fortran77 code written for the model is included in Appendix II. The code will also run on an IBM or compatible personal computer fitted with a math co-processor and graphics capability; a six layer rotated accretion takes about 90 seconds to compute on a regular PC.

**E. Model Applications**

This model is intended for use in the laboratory, in conjunction with wind-tunnel experiments. There the required input data are readily available, as accurately as present equipment and methods can determine them.

However, if this model or some future version of it were to be applied in the field, the situation becomes very different. Good estimates of the input data (duration, wind speed, liquid water content, droplet mvd and type of spectrum) must somehow be obtained. In addition, a field version should be able to predict at least the transition to wet icing, and so the air temperature will also be required as input.

These data are rarely available for a remote site. Local meteorological services may provide estimates or forecasts for duration, wind speed, air temperature, and precipitation, but usually not liquid water content or

| | | |
|---|---|---|
| DROPLET MVD | = 0.0000200M | |
| CYL RADIUS | = 0.0080 | M |
| AIR SPEED | = 12.00 | M / SEC |
| LWC | = 0.00050 | KG / M**3 |
| TIME STEP | = 150.00 | SEC |
| BETA ZERO | = 0.652 | |
| ALPHA MAX | = 82.2 | DEGREES |
| ROTN INCR | = 10.0 | DEGREES |
| SPECTRUM: | SIMULATED | |

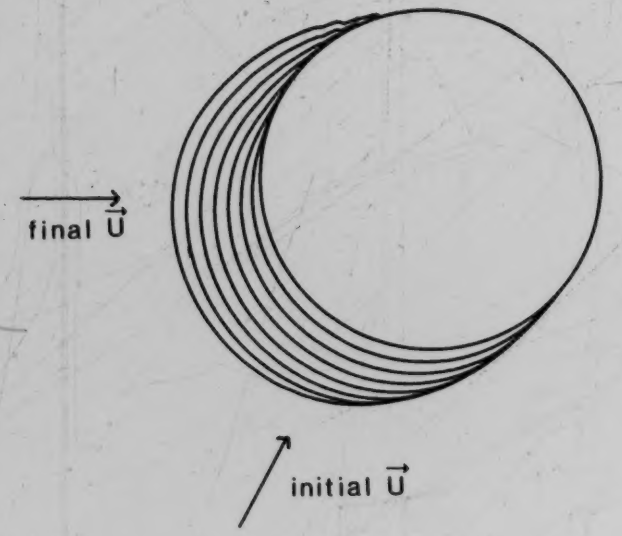



Figure III.4 - An example of the model output for a rotating cylinder.

cloud droplet size.

Clearly, real-time icing forecasting with such a model is a long way off. However, the model could conceivably be used to determine icing climatologies for different regions where adequate meteorological data have been obtained. This would mean sufficiently long-term observations of liquid water contents and droplet sizes (i.e. cloud type) to give an idea of the typical values of these quantities under the typical meteorological situations which lead to icing events in a given region.

However, given the current state of knowledge of field conditions, the best application for this model remains in furthering the understanding of what is happening under controlled rime icing conditions in the laboratory, which it is capable of doing much more efficiently and cheaply than previous models.

## References


Gates, E.M. 1981: FROST Tunnel. Departmental Report No. 26, Department of Mechanical Engineering, The University of Alberta. 10 pp.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories. In: Collected Works of Irving Langmuir, Vol. 10. Pergamon Press, 348-393.

Lozowski, E.P., Finstad, K.J. and Gates, E.M. 1985: Comments on "Calculation of the Impingement of Cloud Droplets by the Finite-Element Method". *J. Atmos. Sci.*, 42, 306-307.

Lozowski, E.P. and Gates, E.M. 1984: Ice Accretion on Structures in a Marine Environment. Proceedings, Third International Symposium on Offshore Mechanics and Arctic Engineering, New Orleans, 8 pp.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Non-Rotating Cylinder Part I: A Simulation Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

Ludlam, F.H. 1951: The Heat Economy of a Rimed Cylinder. *Quart. Jour. Roy. Met. Soc.*, 77, 663-667.

Mason, B.J. 1971: The Physics of Clouds, 2nd Edition. Clarendon Press, p. 93.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Cylinder by the Finite Element Method. *J. Atmos. Sci.*, 38, 1027-1036.

Oleskiw, M. 1982: A Computer Simulation of Icing on Airfoils. Ph.D. Thesis, The University of Alberta, 302 pp.

## IV. Local Densities of Circular Cylinder Rime Accretions

### A. Introduction

A situation very commonly described for the study of rime ice accretion is that of a non-rotating, circular cylinder, enveloped in a steady cross-flow of cold, cloudy air. Rime ice is formed when all of the impinging cloud droplets are able to freeze completely at the point of impact, and before the arrival of another droplet. Rime ice is always 'dry', which means there is no liquid water on the surface or trapped within the ice. The conditions necessary for dry ice to form are described by the so-called "Ludlam limit" (Ludlam, 1951), a maximum value of cloud liquid water content computed as a function of air temperature, cylinder size and air speed. Above this limiting value, the accretion will be 'wet', or glaze ice, where the droplets may run along the surface before freezing, or are incorporated into pockets of liquid water within the ice. This results in a much higher and more uniform density of ice. This study deals only with rime ice.

Examination of a rime accretion in cross-section (Figure IV.1) reveals details of the internal structure which govern the local density. Air bubbles and channels are apparent, usually increasing in number and size toward the lateral edges, away from the stagnation line. At the extreme edges of the accretion are often found the individual elongated or fan-shaped structures known as

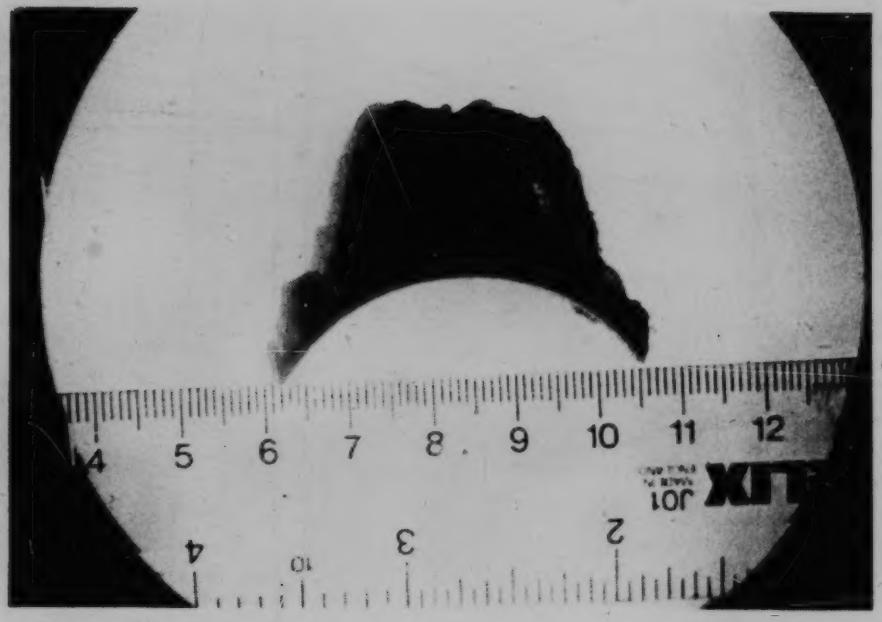



Figure IV.1 - A 1.5 cm thick cross-section of a circular cylinder rime accretion, viewed in transmitted light. The accretion conditions are those of Sample 1 in Table IV.1.

'rime feathers', which seem to merge into the accretion from the sides. These are seen more easily in a lateral view (Figure IV.2).

It is apparent even from such a cursory examination that there should be a trend in the local ice density, decreasing from the stagnation line out toward the edges. When averaged over intervals larger than the typical air channel size, a smooth distribution should emerge. The actual shape of this distribution and how it arises are of interest to modellers of the icing process.

Numerical models have been devised which predict the results of accretion experiments performed in wind tunnels under controlled conditions, e.g. Oleskiw (1982), Lozowski, Stallabrass and Hearty (1983), McComber and Touzot (1981). The parameters which determine the rate of ice growth in such models are the accretion object's size, the air temperature and pressure, air speed, liquid water content and the sizes and relative number distribution of the water droplets in the air stream. These quantities are needed to calculate the local and overall collision efficiencies on the collecting object, which together with the local ice density determines the mass and the shape of the modelled ice deposit.

Although collision efficiency and its local distribution have been well studied, at least for circular cylinders (see for example, Oleskiw, 1982, McComber and Touzot, 1981, Langmuir and Blodgett, 1946), the density of

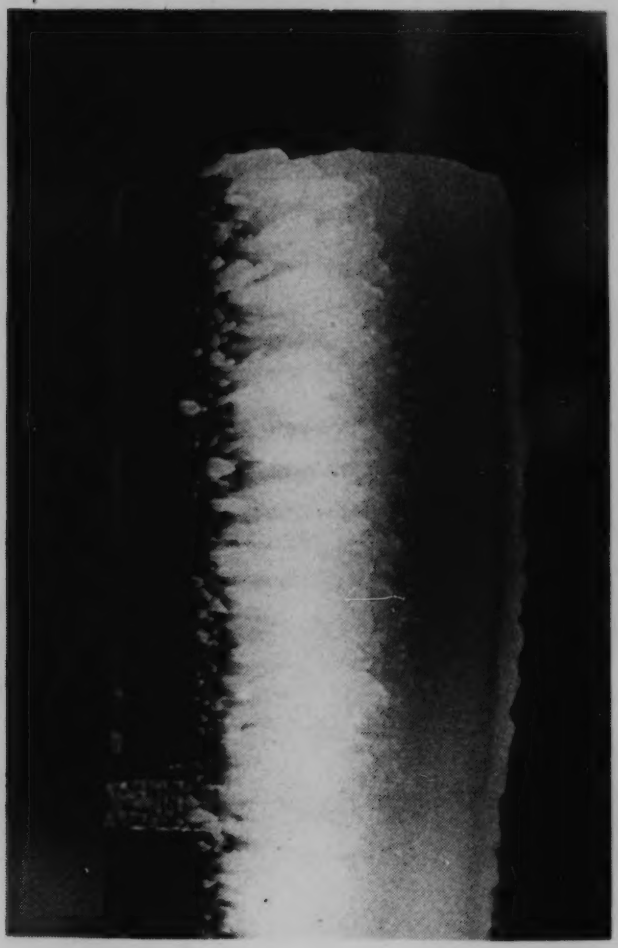


Figure IV.2 - A typical circular cylinder accretion showing rime feathers.

rime has not received as much attention. Some studies have dealt with the densities of ice accretions on rotating cylinders or hailstones (e.g. Macklin, 1962, List, Cantin and Ferland, 1970, Makkonen and Stallabrass, 1984).

Bain and Gayet (1983) have attempted to adapt these results to the fixed cylinder case; their method will be covered in the next section. Problems with this method, very similar to those outlined by Rasmussen and Heymsfield (1985), are also discussed, and a new density distribution is derived directly from experimental accretion shapes. Both distributions are tested in the model, and also against direct laboratory measurements of local density.

## B. Rotating Cylinder Densities Applied to the Non-rotating Case

Macklin (1962) made extensive measurements of the bulk densities of rotating cylinder ice deposits. The results were found to correlate best with the parameter, called R, given by:

$$R = r_d V_0 / T_s \qquad \text{IV.1}$$

where $r_d$ is the radius of the median volume droplet, $V_0$ is the impact speed of that droplet at the stagnation line, and $T_s$ is the mean surface temperature of the accretion. Similar correlations were found for the experimental results of Bain and Gayet (1983) and Makkonen and Stallabrass (1984)

on rotating cylinders, although the latter found significantly larger densities in the lower range of R.

Makkonen and Stallabrass attribute this difference to their method of measuring droplet sizes with optical probes. Macklin's droplets were measured using the oil slide method, which according to Makkonen and Stallabrass gives underestimated sizes, in comparison with the optical probe method. Makkonen and Stallabrass' version of the correlation for the range $0.2 < R \leq 170$ is:

$$\rho(R) = 0.378 + 0.425(\log_{10}(R)) - 0.0823(\log_{10}(R))^2. \qquad \text{IV.2}$$

Bain and Gayet (1983) made an attempt to apply the earlier correlation of Macklin to densities on a fixed cylinder accretion, by introducing local variations into the R parameter as follows:

$$R(\theta) = r\, V_r(\theta) \,/\, T_s(\theta), \qquad \text{IV.3}$$

where now $V_r(\theta)$ is the radial component of droplet impact velocity as a function of the radius angle $\theta$ (measured with respect to the stagnation line). Bain and Gayet derive the total droplet speed at impact from an assumed cosine distribution of speed with cylinder radius angle, scaled to the stagnation line value as calculated by Langmuir and Blodgett. $T_s(\theta)$ is the variation of local surface temperature calculated from the model of Lozowski,

Stallabrass and Hearty (1983).

An initial objection to this approach is philosophical. Macklin's correlation is a purely empirical one; it is not a physical model and therefore should not be extended to a physical situation so different from that which provided the original data.

On the fixed cylinder, $V_r(\theta)$ is not even analogous to $V_0$ on the rotating cylinder, which is the maximum impact speed felt at any one point during one rotation. Furthermore, on the rotating cylinder, the droplets arrive at a given point from a large range of angles, which should perhaps result in closer packing of the droplets and a higher density. More importantly, the shadowing effects of feather structures and the formation of air channels between them is not accounted for.

However, a more concrete objection is found by comparing this type of distribution to that found by direct measurement of the density, and by comparing the model accretion shapes and masses it produces to experimental profiles. The first comparison will be discussed in Section D of this Chapter. For the model comparison the simple rime accretion model described in Chapter III is used.

The experimental accretion profile shown on the left in Figure IV.3 was grown, under the conditions listed, in the University of Alberta FROST tunnel. Also shown are model profiles produced by three different distributions of local

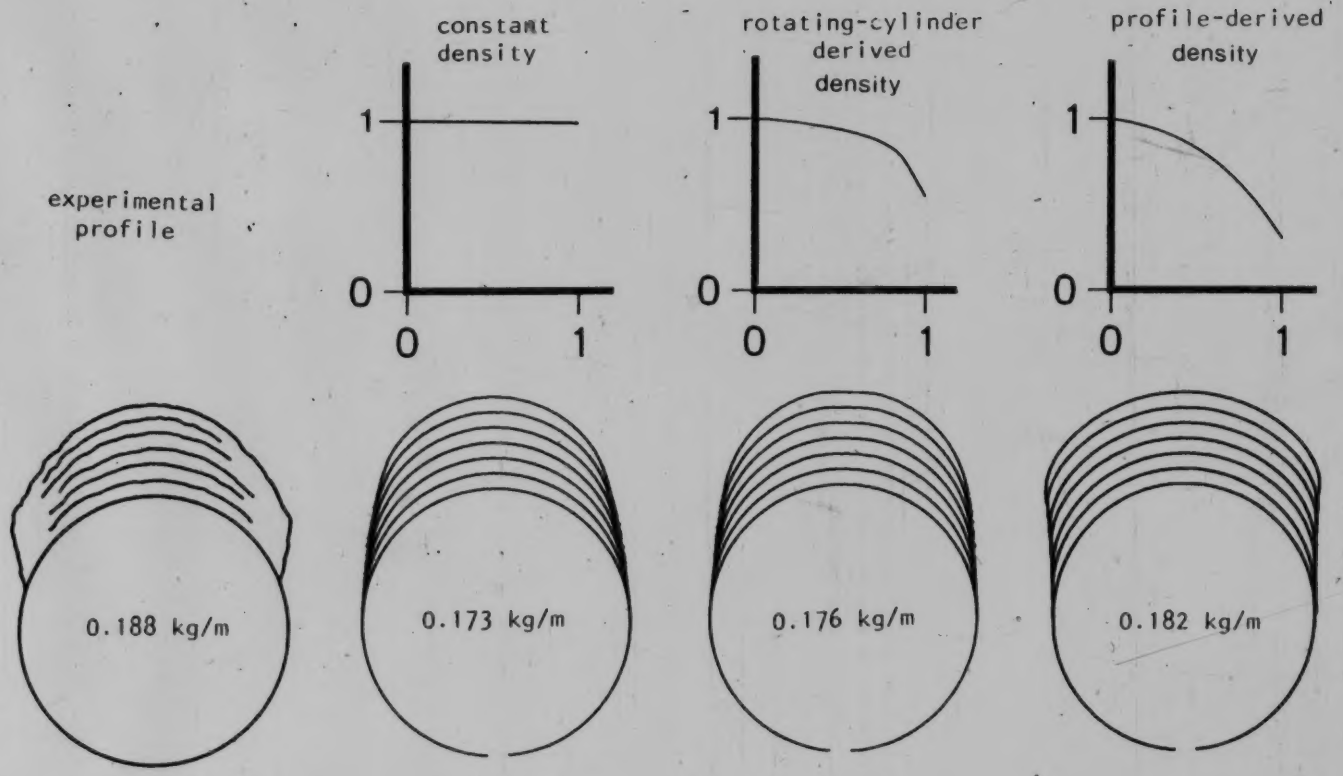



Figure IV.3 - Experimental and three model accretions showing the effects of three different density distributions (plotted above as $\rho/\rho_0$ vs. $\alpha/\alpha_{max}$). Conditions are: $D_d$ = 51 $\mu$m, $D_c$ = 0.0254 m, U = 10.7 m/sec, $lwc \cong 0.49$ g/m$^3$, $\tau$ = 300 sec.

density. The profile at centre left results from a constant density, that at centre right from a distribution that is derived following Bain and Gayet. The profile at the far right uses a third distribution which is described below.

The three density distributions are illustrated above their respective profiles. For easy comparison, each is plotted as $\rho/\rho_0$ vs. $\alpha/\alpha_m$, where $\rho_0$ is the local density at the stagnation line, $\alpha$ is the local angle of the surface normal with respect to the free stream direction, and $\alpha_m$ is defined in Chapter III. On the bare cylinder, $\alpha$ is identical with $\theta$, but not for subsequent ice layers. The angle $\alpha$ is used rather than $\theta$, so that the density distribution will be time-dependent, changing as the ice shape changes.

The middle distribution is derived from Equation IV.1, using $V_r(\alpha)$ as calculated by Bain and Gayet, and $T_s(\alpha)$ calculated by the model of Lozowski, Stallabrass and Hearty. The resulting curve is fitted with a third-order polynomial:

$$\rho(\alpha/\alpha_m) \; / \; \rho_0 = (1 - 0.303 \; (\alpha/\alpha_m) \qquad \text{IV.4}$$

$$+ 0.936 \; (\alpha/\alpha_m)^2 - 1.074 \; (\alpha/\alpha_m)^3).$$

The second modelled profile is very similar to the first, constant density profile, which was too narrow in comparison with the experimental shape. Both of the modelled masses are less than the observed mass, by 8 and 6

percent, respectively. The result using the third distribution shown is in better agreement with the experimental shape, by having a widening profile, and a mass within 3 percent of observation. This distribution and its derivation are described next.

### C. Profile-Derived Density Distribution

The thickness (Th) of ice deposited in time $\tau$ at a given point on the bare cylinder surface, which we describe by its surface angle $\alpha$, is

$$Th(\alpha) = \beta(\alpha)\ U\ lwc\ \tau\ /\ \rho(\alpha), \qquad IV.5$$

where U is the free-stream speed, and lwc is the liquid water content. This version of the dry-growth equation ignores curvature effects, e.g. as described by Lozowski, Stallabrass and Hearty (1983). This is a reasonable simplification given the very small spacing of the points which define the icing surface in the model.

Given IV.5 then, if both the local thickness and the local collision efficiency $\beta(\alpha)$ are known, and if the interval $\tau$ is assumed to be small enough that $\beta(\alpha)$ and the density $\rho(\alpha)$ remain essentially constant, IV.5 may be solved for $\rho(\alpha)$.

Figure IV.4 shows another experimental accretion grown in the FROST tunnel, and its growth conditions. The total accretion time was 30 minutes. Every five minutes, a layer

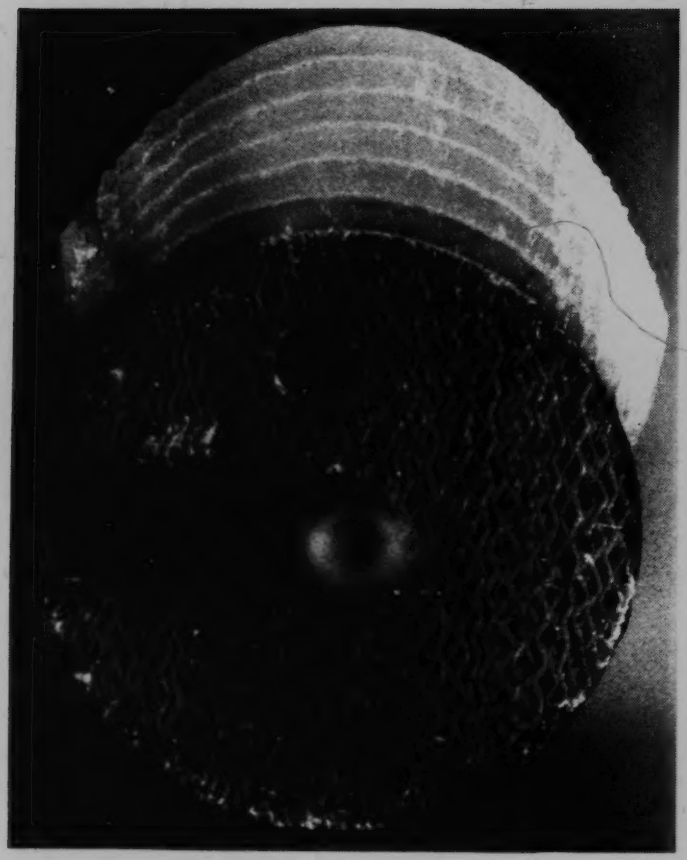


Figure IV.4 - A circular cylinder rime deposit in profile, with paint layers marking the surface at five minute intervals. Conditions are those of Case B in Table III.1.

of paint was sprayed onto the surface, allowing changes in the shape of the ice surface to be followed with time. It also allowed local thickness measurements of thin layers to be obtained, which are unaffected by the melting or sublimation losses that must occur on the top surface, between the time of the end of the test and photography of the profile.

The distribution of local collision efficiency was calculated from the median volume diameter, and an assumed droplet size distribution, scaled to the mvd, based on measurements taken in the same wind tunnel. The mvd was originally measured by the oil slide method, which is known from the work of Makkonen and Stallabrass (1984) to be an overestimate; but for Case B of Table IV.1 the oil slide estimate falls outside of the size range for which Makkonen and Stallabrass' empirical correction formula is valid. Therefore, an mvd has been adopted which was derived from the stagnation line thickness measurement, the liquid water content measurement, and an assumed local density for the stagnation line of 890 kg / $m^3$ (a reasonable density in this case based on visual inspection of the accretion). The method used to derive this estimate of the mvd is described in Appendix III.

For Case A of the same table it was possible to make use of the correction formula mentioned above to correct the oil slide measured mvd.

In this way the distribution $\rho(\alpha)$ has been found for several layers of two different accretions. The data are listed in Table IV.1 and plotted in Figure IV.5. A cubic polynomial fit (solid curve, Figure IV.5) gives:

$$\rho(\alpha/\alpha_m) / \rho_0 = (1 - 0.143\ (\alpha/\alpha_m) - 0.246\ (\alpha/\alpha_m)^2 - 0.309\ (\alpha/\alpha_m)^3) \qquad \text{IV.6}$$

The relative distribution is thus assumed independent of lwc and air temperature; this assumption implies that these factors only come into the determination of the scaling parameter $\rho_0$. Normalization of the distribution by the two parameters $\rho_0$ and $\alpha_m$ makes the expression above easy to use in an icing model, assuming that it is generally applicable to a range of accretion conditions. Evidence for this was presented in Figure III.3 of the last Chapter, where the model results are compared to several independent wind tunnel experiments.

**D. Laboratory Measurements of Local Density**

For further evidence to support our proposed distribution of local density, a series of direct measurements were made from wind-tunnel accretions. Following the techniques described by List, Cantin and Ferland (1970), local densities were measured by drilling small holes (2.5 - 3.0mm diameter, less than 1 cm depth) in

| | $\alpha$ (degrees) | Th (mm) | $\beta(\alpha)$ | $\rho$ (kg/m$^3$) |
|---|---|---|---|---|
| Case A | 0 | 3.34 | 0.407 | 767 |
| | 10 | 3.12 | 0.393 | 792 |
| | 20 | 3.25 | 0.334 | 646 |
| | 30 | 2.9 | 0.249 | 540 |
| | 40 | 1.7 | 0.155 | 574 |
| | 50 | 0.9 | 0.069 | 482 |
| Case B | 0 | 1.48 | 0.842 | 890 |
| | 10 | 1.48 | 0.825 | 871 |
| | 20 | 1.48 | 0.776 | 820 |
| | 30 | 1.40 | 0.699 | 780 |
| | 40 | 1.33 | 0.592 | 696 |
| | 50 | 1.03 | 0.458 | 695 |
| | 60 | 0.81 | 0.302 | 583 |

Table IV.1 - Local densities as a function of surface slope derived from thickness measurements of shallow ice layers accreted at -10°C under the following conditions:

| | $D_d$ ($\mu$m) | $D_c$ (m) | lwc (g/m$^3$) | U (m/sec) | $\tau$ (sec) |
|---|---|---|---|---|---|
| Case A | 15.5 | 0.0254 | 0.98 | 10.7 | 600 |
| Case B | 51.0 | 0.0254 | 0.49 | 10.7 | 300 |

$\beta(\alpha)$ is calculated from Equations II.11, III.3 and III.4.

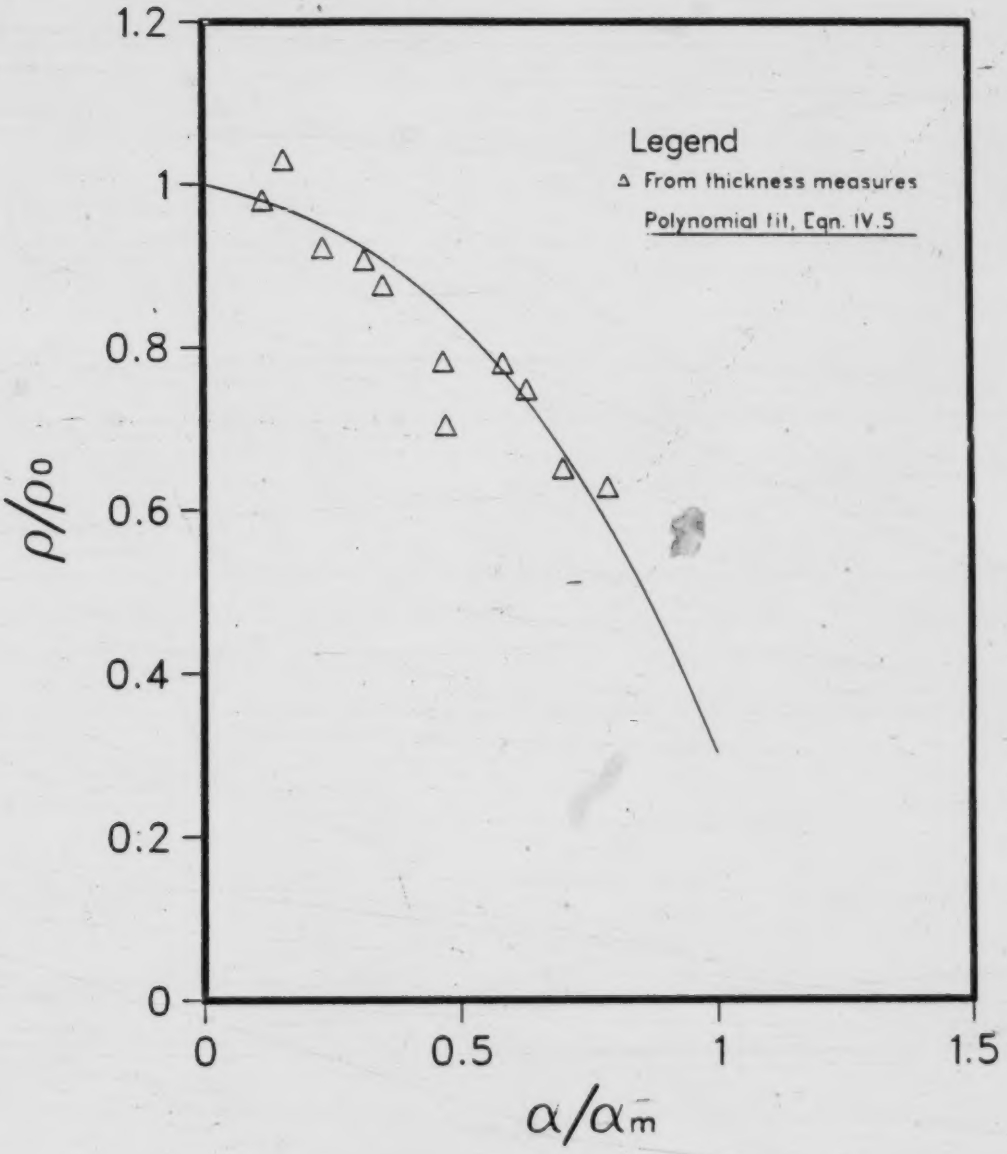



**Figure IV.5 - Local densities derived from thickness measurements of shallow ice layers from Table IV.1. The solid curve is a polynomial fit to the data (Equation IV.3).**

cross sectional cuts from two different rime accretions (see Figure IV.6). The location, volume and mass loss were recorded for each hole. The mass was measured to within 0.00005 g with a Mettler model H20 balance. For the data of Sample 1, Table IV.2, the hole depths and diameters from three sections of a single accretion were measured to within .05 mm with a Vernier caliper. For Sample 2 of Table IV.2, holes from two sections of accretions grown under nearly identical conditions were measured to 0.01mm in depth and 0.001mm in diameter, with the same bore and depth gauges used by List et al. (1970) to determine local densities in hailstones. The surface angles were approximated by the more easily measured radius angles, since paint layers to define the intermediate surfaces would have altered the densities. This is a good assumption for shallow layers, at any rate, and measurements have been used only from the layer immediately above the cylinder surface. The error in using the radius angle approximation is estimated to be ±five degrees.

Another source of error is mass loss from the entire sample through sublimation and handling during the hole drilling procedure. Several experimental estimates were made of the size of this error, and the average (0.001 g) was subtracted from the measured mass losses from the holes.

The resulting density measurements are shown in Table IV.2 and in Figures IV.7 and IV.8. The solid and dashed curves shown in each of these figures are from Equations

Figure IV.6 - An accretion section of about 1.5 cm. thickness, showing the holes drilled for the local density measurements. The condition are those of Sample 1 in Table IV.2

| Sample 1 | | Sample 2 | |
|---|---|---|---|
| $\alpha$ (±5°) | $\rho$ (±60 kg/m$^3$) | $\alpha$ (±5°) | $\rho$ (±30 kg/m$^3$) |
| 2 | 786 | 2 | 907 |
| 2 | 667 | 3 | 902 |
| 2 | 660 | 4 | 723 |
| 6 | 719 | 6 | 747 |
| 8 | 620 | 7 | 770 |
| 8 | 792 | 8 | 909 |
| 10 | 714 | 13 | 788 |
| 12 | 645 | 15 | 815 |
| 12 | 688 | 17 | 766 |
| 12 | 685 | 19 | 724 |
| 15 | 629 | 23 | 823 |
| 18 | 519 | 24 | 683 |
| 20 | 715 | | |
| 21 | 630 | | |
| 27 | 675 | | |
| 28 | 427 | | |
| 30 | 681 | | |
| 30 | 733 | | |
| 30 | 701 | | |
| 35 | 539 | | |
| 38 | 580 | | |
| 42 | 749 | | |
| 50 | 504 | | |

Table IV.2 - Experimentally measured local densities as a function of surface slope taken from accretions grown under the following conditions:

| $D_d$ (m) | $D_c$ (m) | lwc (g/m$^3$) | U (m/sec) | T (°C) |
|---|---|---|---|---|
| 16.4 | 0.046 | 0.53 | 14.9 | -14 |
| 14.0 | 0.0254 | 0.36 | 30.0 | -10 |

IV.2 and IV.4, respectively, where the choice of $\rho_0$ has been based on the density measurements, and $\alpha_{max}$ is calculated from the median volume diameter droplet, following the methods of Chapters II and III.

The scatter of points in both figures is considerable, and is much larger than the measurement errors represented by the error bars will allow. The most likely cause of the large scatter is in the nature of the rime itself. The internal air channels are scattered at random, and since they are often of the same size as the holes, this will introduce a natural scatter. Also, in the very fragile rime, the drilling process may cause walls between adjacent air channels to collapse, resulting in additional mass loss. This problem is especially bothersome near the edges of the accretion, where the density is lowest. Here, walls between the holes themselves will sometimes collapse, making it very difficult to obtain density values at large surface angles. Part of the problem might be alleviated by using accretions samples grown on a larger diameter cylinder, so that the holes do not have to be drilled so close together to get the same interval of surface angle. Unfortunately, the only accretions available were grown on cylinders of one and two inches in diameter.

For whatever reason, the results are too scattered to allow a clear choice between one parameterized distribution and the other. This is due, in part, to the lack of data at

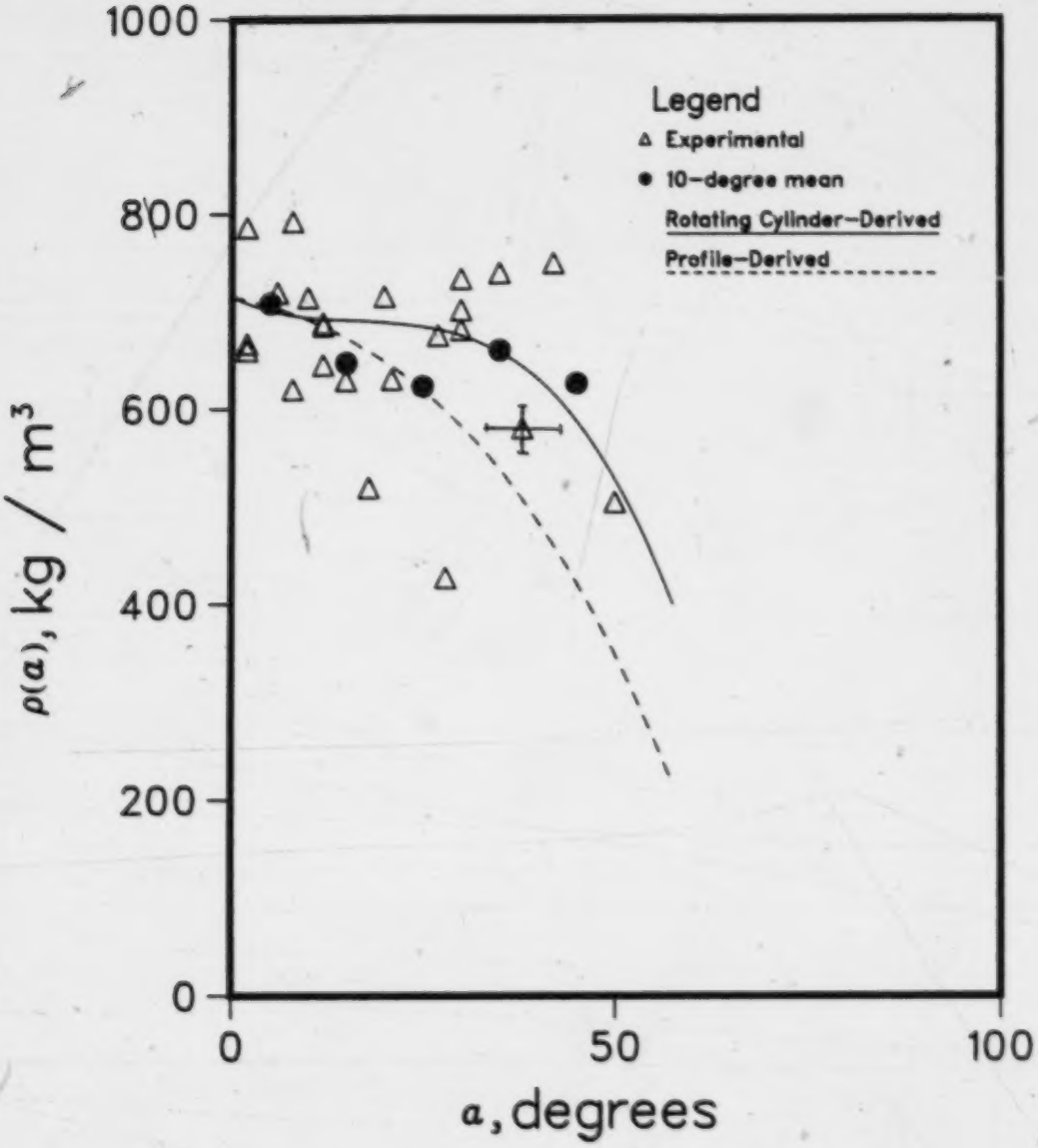



Figure IV.7 - Local density vs. surface angle; experimental measures and parameterized curves for the conditions of Sample 1 in Table IV.2. The filled symbols represent mean values for 10° intervals in surface slope.

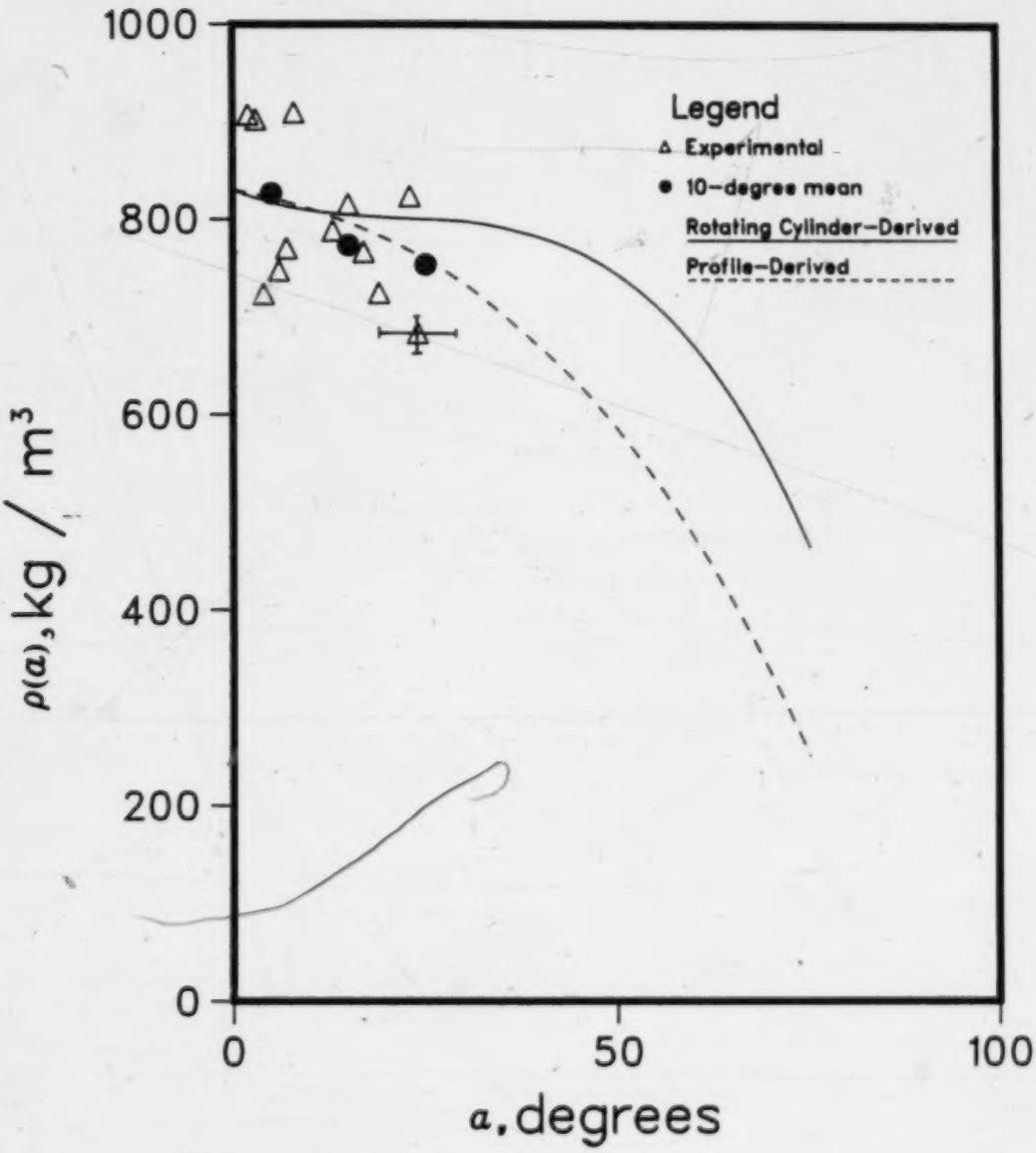



Figure IV.8 - Local density vs. surface angle; experimental measures and parameterized curves for the conditions of Sample 2 in Table IV.2. The filled symbols represent mean values for 10° intervals in surface angle.

large surface angles. When averaged over 10 degree intervals (filled symbols) the experimental data in both figures favour the profile-derived curve up to a surface angle of 30°, but for larger angles in Figure IV.7 they lie closer to the rotating cylinder-derived curve. However, there are far fewer points defining the averages for these larger angles, and none at all for Figure IV.8.

Although these direct density measurements have not been conclusive, the model results and the practical advantages of the profile-derived distribution justify its adoption over the other. The application of this method is efficient and inexpensive, assuming that the scaling parameters may be easily determined. One of these parameters, the maximum accretion angle, $\alpha_{max}$, may be quickly approximated according to the analytical function given in Chapter II, but $\rho_0$ remains to be determined.

**E. Local Density at the Stagnation Line**

Qualitative observations of many rime accretions show that the stagnation line density also varies with the environmental conditions. Since there are, at present, insufficient quantitative data on this variation, the correlation for rotating cylinder densities, Equation IV.1, may perhaps be applied. The justification for doing so here, but not for the entire density distribution, is that there is no shadowing at the stagnation line, and the conditions are more nearly analogous to the rotating

cylinder case.

However, some important differences must be noted between the two situations. In the rotating cylinder correlation, the parameter $V_0$ is the impact speed of the median volume droplet at the stagnation line only. Over the remainder of the accretion surface, droplets impact at speeds ranging between zero and $V_0$, so the average impact speed, at least for cases with large $\alpha_{max}$, is roughly one half $V_0$, and perhaps somewhat less if $\alpha_{max}$ is small. Thus, $V_0$ for the rotating cylinder is $\leq$ twice the average speed of impact. On the stagnation line of the non-rotating cylinder, the impact speed is constant and equal to $V_0$, so in order to substitute into the R parameter the correct analogy for twice the average speed, we use twice $V_0$. $V_0$ is estimated from Equation II.11.

The R parameter is further altered by substituting for the mean surface temperature, the value at the stagnation line ($T_{s,0}$) which is determined using the model of Lozowski, Stallabrass and Hearty (1983). The new correlation parameter is then:

$$R' = 2\, r_d\, V_0 \,/\, T_{s,0} \qquad \text{IV.7}$$

The results are shown in Figure IV.9, for five accretions for which $\rho_0$ has been experimentally measured by the methods described in the last section. The fit to Equation IV.1 is quite good, considering the size of the

measurement errors in $\rho_0$ (i.e., the same size as for Figure IV.8). Until more experimental data are available, this is an acceptable method for estimating $\rho_0$.

**F. Summary**

Previous methods for estimating the local variation of rime ice density are examined, and found to be inadequate. A new method is derived empirically, and tested in model accretions, and against direct laboratory measurements of local density. The laboratory measured distribution is inconclusive, but model results show the new method to be more successful than the old for predicting the shapes and masses of circular cylinder rime ice accretions.

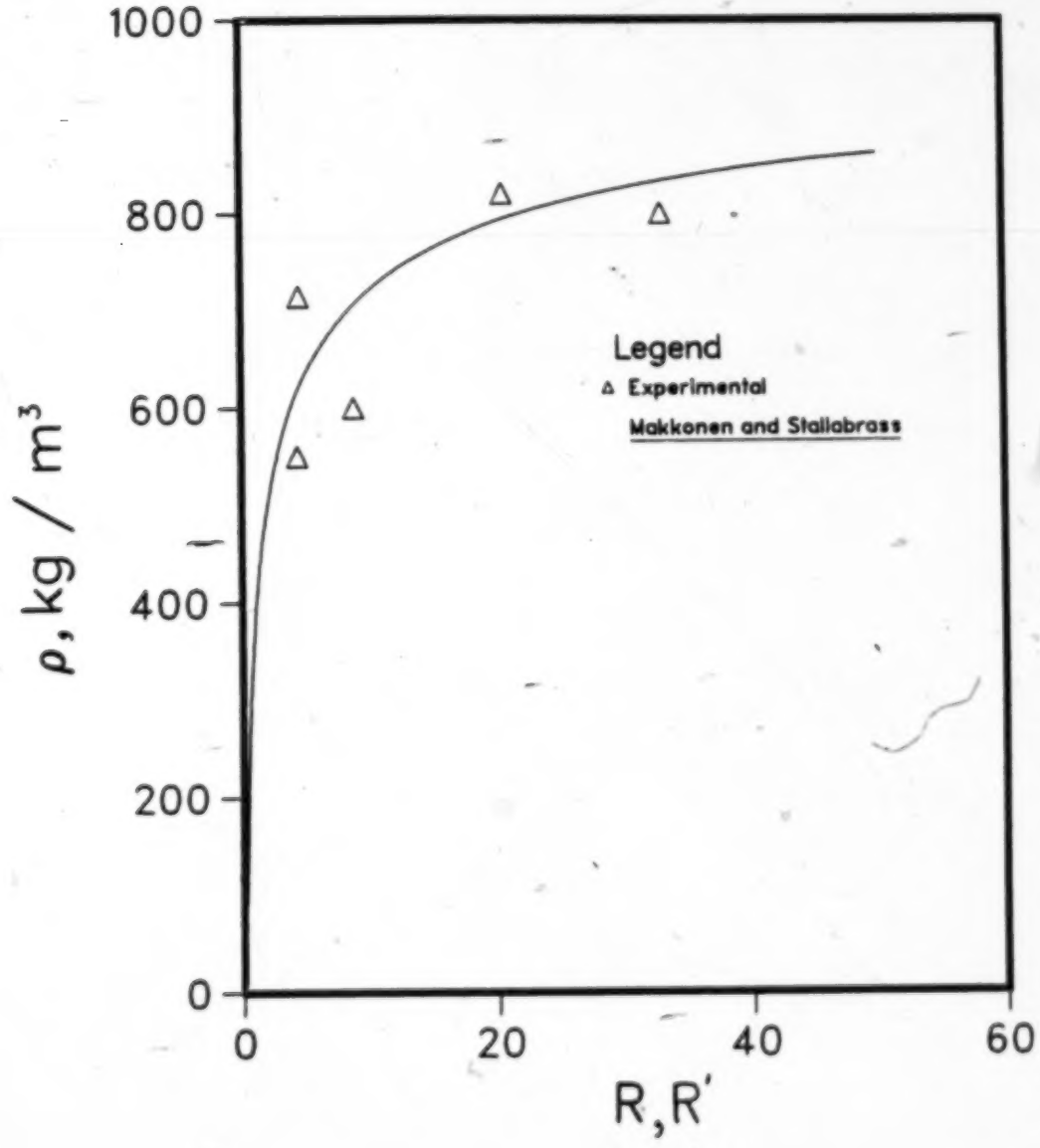



Figure IV.9 - Stagnation line density measurements vs. R' (symbols), and the rotating cylinder density correlation of Makkonen and Stallabrass, Equation IV.2, vs. R (solid line). R and R' are defined in Equations IV.1 and IV.7, respectively.

## References


Bain, M. and Gayet, J.F. 1983: Etude de L'accretion de la Glace Atmospherique II. Modelisation Numerique avec Parametrisation de la Variation de la Mass Specifique du Givre en Fonction de la Position Angulaire sur un Cylindre. *J. Rech. Atmos.*, 17, 299-311.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories. Collected Works of Irving Langmuir, Vol 10. Pergamon Press, 348-393.

List, R. Cantin, J.-G. and Ferland, M. 1970: Structural Properties of Two Hailstone Samples. *J. Atmos. Sci.*, 27, 1080-1090.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Rotating Cylinder Part I: A Simulation Mod Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

Ludlam, F. H. 1951: The Heat Economy of a Rimed Cylinder. *Quart. J. Roy. Meteor. Soc.*, 77, 663-666.

Macklin, W.C. 1962: The Density and Structure of Ice Formed

by Accretion. *Quart. J. Roy. Meteor. Soc.*, 88, 30-50.

Makkonen, L. and Stallabrass, J. 1984: Ice Accretion on Cylinders and Wires. National Research Council Technical Report TR-LT-005, NRC No. 23649, 43 pp.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Cylinder by the Finite Element Method. *J. Atmos. Sci.* 38, 1027-1036.

Rasmussen, R.M. and Heymsfield, A.J. 1985: A Generalized Form for Impact Velocities Used to Determine Graupel Accretional Densities. *J. Atmos. Sci.* 42, 2275-2279.

## V. Rime Icing on Airfoils

### A. Introduction

A numerical simulation of ice accretion on an arbitrary airfoil section involves greater computational difficulty than does the corresponding problem of icing on circular cylinders. The major reason is the requirement for numerical integration of the impinging droplet trajectories in order to determine collision efficiencies. Unlike the cylinder case, there is (except for the Joukowski airfoil) no analytical solution of the potential flow to simplify the task, nor are there convenient tables of previously calculated collision efficiencies as functions of the flow and inertia parameters.

This requirement adds considerably to the model's consumption of computing time and resources. Previous airfoil accretion models (e.g. Oleskiw, 1982, MacArthur et al., 1982, Cansdale and Gent, 1983) have thus tended to be expensive to run, and may be prone to numerical instabilities.

However, from the results of one such model (Oleskiw 1982), simple parameterizations have been derived for the local collision efficiency distribution with surface slope on NACA 0015 and 0012 airfoils. Three separate analytical functions approximate this distribution fr monodisperse or average droplet spectra, or for any spectrum when the angle of attack is non-zero. Use of these functions requires the

determination of only one value of local collision efficiency, that at the stagnation line; thus considerably reducing the computation time. This single value may then be re-determined at intervals during the accretion to give the model a true time-dependence.

These parameterizations have been incorporated into a new, more efficient numerical icing model. Other computational simplifications include the representation of the airfoil and ice surfaces by straight line segments rather than by a spline curve, and a straightforward algebraic method to determine the droplet's point of collision with the surface. An approximate distribution function for the local density has also been introduced. Again, rime icing only is considered, i.e. the air temperature is assumed to be below -10° C, and the impinging water is assumed to freeze immediately and completely. Most of the parameterizations and simplifications could also be used in a model for wet accretion.

The model results are presented in comparison with those of Oleskiw (1982) and with real accretions grown in wind tunnel experiments.

**B. Model Outline**

A brief outline of the model algorithm is as follows:

**Step 1.** Specify the x, y coordinates of the original airfoil surface. The coordinates are dimensional,

with the origin at the nose of the airfoil.

**Step 2.** Find the local collision efficiency at the stagnation line, $\beta_0$, by the following procedure:

1. Solve for the potential flow field about the airfoil or accretion shape by the method of Kennedy and Marsden (1976).
2. Integrate two droplet trajectories near the stagnation line and find their collision points.
3. Calculate $\beta_0$ from $\Delta y / \Delta \ell$, the ratio of initial separation normal to the free-stream of the trajectories to their final separation in arc length on the surface.

**Step 3.** For each surface point, calculate the thickness accreted perpendicular to the surface during one layer time step $\tau$. The local collision efficiency at each point is approximated from one of three parameterized functions of $\beta_0$ and the local surface slope, $\alpha$, depending on the type of droplet size spectrum and the angle of attack, $\Phi$ (defined as the angle between the free-stream direction and the airfoil chord). The local density distribution is also approximated from a parameterized function of surface slope, $\alpha$.

**Step 4.** Calculate the mass and mean density of the accreted ice layer.

**Step 5.** Repeat Steps 3 and 4 for the number of layers requested. Repeat Step 2 only for every fifth layer. The layer time step, $\tau$, should be chosen so that the combined stagnation line thickness of five layers is not more than 1/10 of the original chord length.

**Step 7.** Plot the original airfoil and all accreted layers.

The sections below will discuss each of Steps 1 through 4 in detail.

**C. Specifying the Airfoil Shape**

This version of the model employs the standard airfoil section known as the NACA (United States National Advisory Committee on Aeronautics) four-digit wing section, for which Abbott and von Doenhoff (1959) give the thickness distribution in non-dimensional x,y-coordinates normalized to the chord length, as:

$$y = 0.05\, h_0\, (0.2969\, x^{0.5} - 0.126\, x - 0.3516\, x^2 + 0.2843\, x^3 - 0.1015\, x^4) \qquad \text{V.1}$$

The number $h_0$ gives the maximum airfoil thickness as a

percentage of the chord length, as shown in Figure V.1, and also makes up the last two digits of the "four-digit" designation. The first two digits represent $c_m$, the maximum camber as a percentage of the chord, and $c_p$, the abscissa of the camber ordinate in tenths of the chord. All of the paramaterizations used in the model are derived assuming that both $c_m$ and $c_p$ are zero. The maximum thickness $h_0$ is entered as model input, but expressed as a fraction of the chord, rather than a percentage.

Although defined above for non-dimensional coordinates, the model carries the airfoil and accretion profiles in dimensional coordinates throughout most of the calculations, because the chord length (initial chord plus accumulated ice thickness) is changing with every layer step. Where a non-dimensional profile is required for the potential flow calculation, the coordinates are normalized using the current chord length.

Since the profile is always represented by a vector of coordinate points, the points must be of sufficiently high density to ensure that the curved surfaces are well approximated by straight line segments joining them. This avoids the expensive computational process of fitting a few surface points with a spline curve, which may be prone to instabilities. It also simplifies the determination of trajectory impact points, but it it does require more data storage. However, even this can be minimized by using a point distribution scheme which concentrates points in the

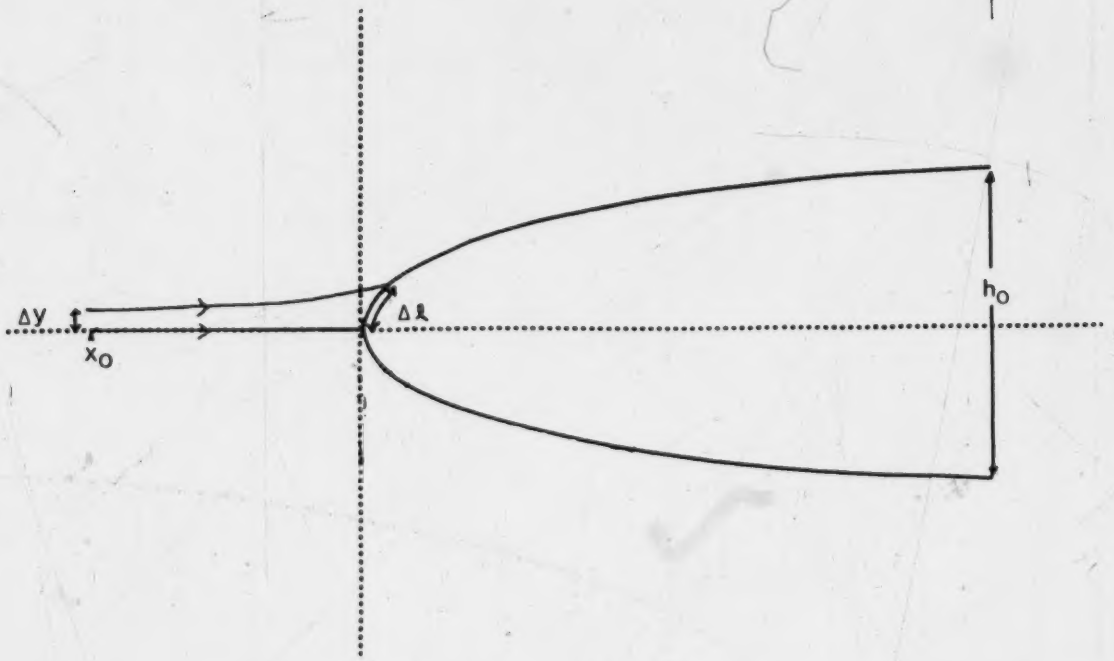



Figure V.1 - Determination of collision efficiency on an airfoil. $\beta = dy/d\ell \simeq \Delta y/\Delta\ell$. Also illustrated is the definition of $\Phi$, the angle of atack.

regions of strong curvature at the nose and tail.

One such scheme is that suggested by Kennedy and Marsden (1976), in which the distribution of non-dimensional x-coordinates is given by:

$$x_i = 0.5\ (1 - \cos(2\pi i\ /\ N)) \qquad \text{V.2}$$

where i lies between 1 and N, an even integer. This scheme was found to give good results with N = 600 to specify the complete airfoil section for plotting purposes, although only half this number of points, defining the front portion of the airfoil, is allowed to accrete ice.

In the potential flow solution, a profile consisting of a much smaller number of points is desired to reduce the computational requirements; Kennedy and Marsden suggest that 40 points is an optimum number. This can be simply constructed from either the original airfoil profile, or from subsequent accretion shapes by taking every 15th point of the denser profile. The new sparse profile will also retain the desired distribution pattern of points.

Accretion thicknesses are calculated for each surface point in the direction perpendicular to the local surface slope. This causes the point separation to increase near the nose with each succeeding layer. After several layers, the separation may have become too large for a reasonable representation of the surface by straight line segments, and has destroyed the original point distribution.

A way to remedy the situation is to shift each coordinate point, after each layer is accreted, closer to the nose along the straight line segment joining it and its nearest neighbour (whose position has already been shifted). The amount of the shift must be carefully chosen to preserve the desired distribution of points, and an accurate profile. Because the original distribution concentrated points near the nose, and because points near the nose diverge faster than elsewhere, those points must be shifted by larger amounts (relative to the segment length) than points further away.

A shift of half the segment length at the nose, decreasing linearly to one eighth at about half the chord length gives the desired results. To test that profiles produced by this scheme are accurate, model profiles were also made with four times the optimum point density. There was no noticeable difference in the resulting plotted profiles.

## D. Local Collision Efficiency

Parameterization schemes were derived in Chapter III for local collision efficiency $\beta$ on circular cylinders as a scaled cosine function of local surface slope $\alpha$. If the numerical results of Oleskiw (1982) for local collision efficiency on a NACA airfoil at 0° angle of attack are plotted against surface slope, they also closely follow a cosine curve, but one which is displaced negatively along

the abscissa. The form holds for several cases computed by Oleskiw, as seen in Figures V.2, V.3 and V.4, and is described by:

$$\beta(\alpha) = \cos(\alpha) - (1 - \beta_0), \qquad \text{V.3}$$

where $\beta_0$ is the value of $\beta$ at the stagnation line. This is the only parameter required to approximate the entire distribution, since the maximum accretion angle $\alpha_{max}$, in radians, follows from:

$$\alpha_{max} = \arccos(1 - \beta_0). \qquad \text{V.4}$$

Appendix IV gives a geometrical derivation of Equation V.3.

So far, a monodisperse droplet population has been assumed. In practice, of course, a spectrum of droplet sizes is present. Often the collision efficiency of the entire population is modelled by a single droplet size at the median volume diameter of the spectrum. This is generally a good approximation to the weighted average $\beta(\alpha)$ distribution calculated according to the relative volumes of the droplet sizes in the spectrum, as discussed in Appendix VI.

In real populations, when relatively large numbers of very small droplets are present in the population, they can change the shape of the $\beta$ distribution, as well as force the mvd-based maximum accretion limit to a smaller value

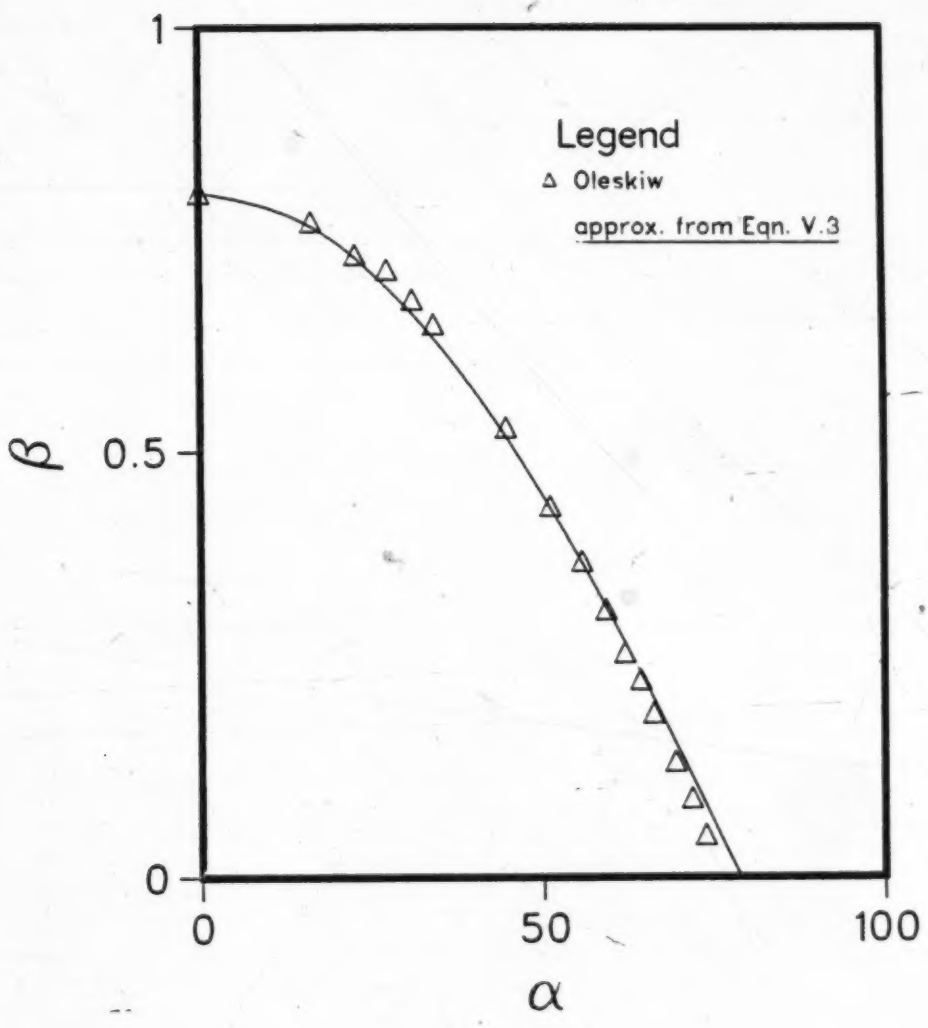



Figure V.2 - Comparison of Oleskiw's numerical results and Equation V.3, for collision efficiency of a single drop size. Airfoil is a NACA 0015, U = 61 m/sec, Chord = 0.213 m, $D_d$ = 20 μm.

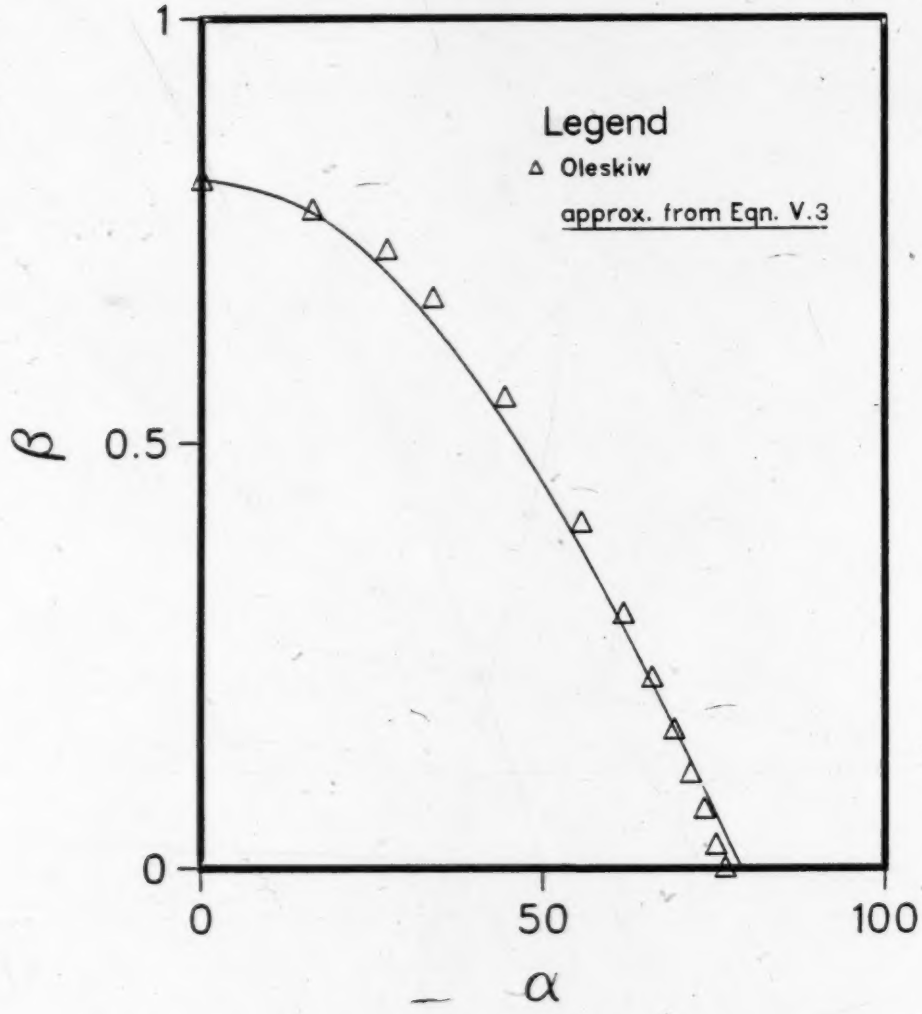



Figure V.3 - Comparison of Oleskiw's numerical results and Equation V.3, for collision efficiency of a single drop size. Airfoil is a Joukowski 15%, U = 78.2 m/sec, Chord = 0.33 m, $D_d$ = 25.5 μm.

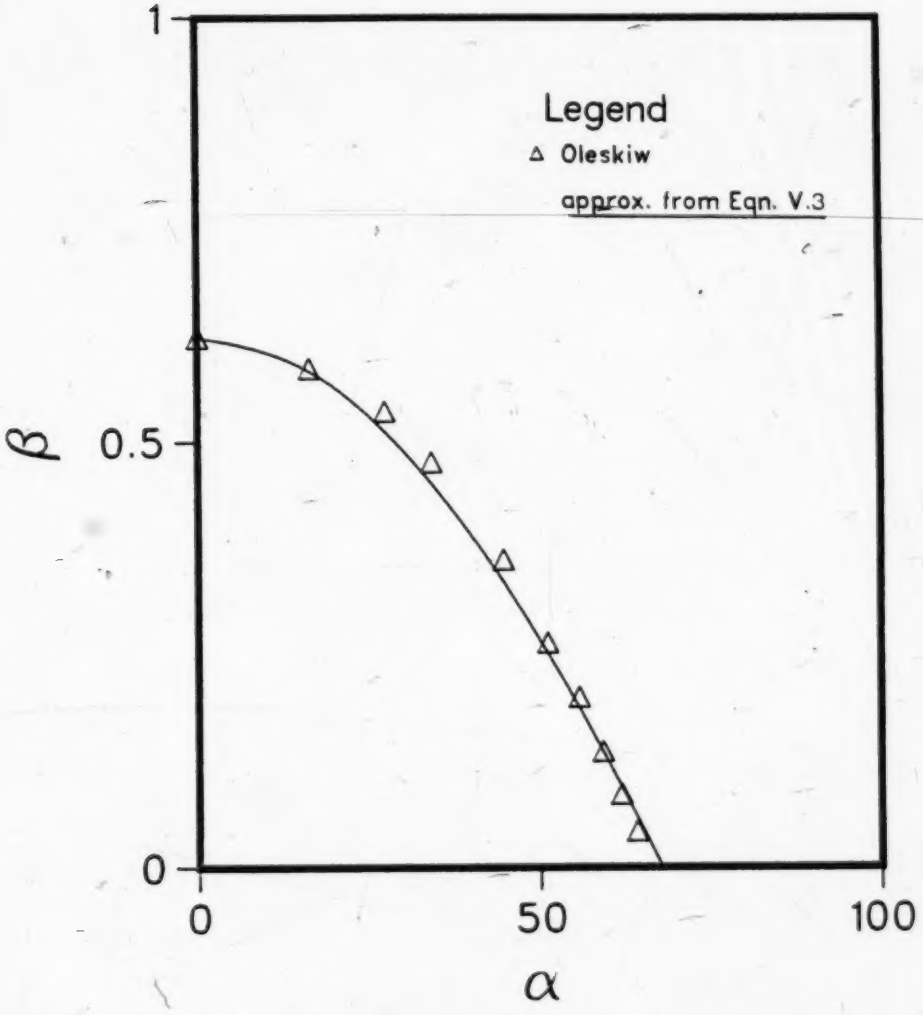



Figure V.4 - Comparison of Oleskiw's numerical results and Equation V.3, for collision efficiency of a single drop size. Airfoil is a Joukowski 15%, U = 78.2 m/sec, Chord = 0.33 m, $D_d$ = 13.2 μm.

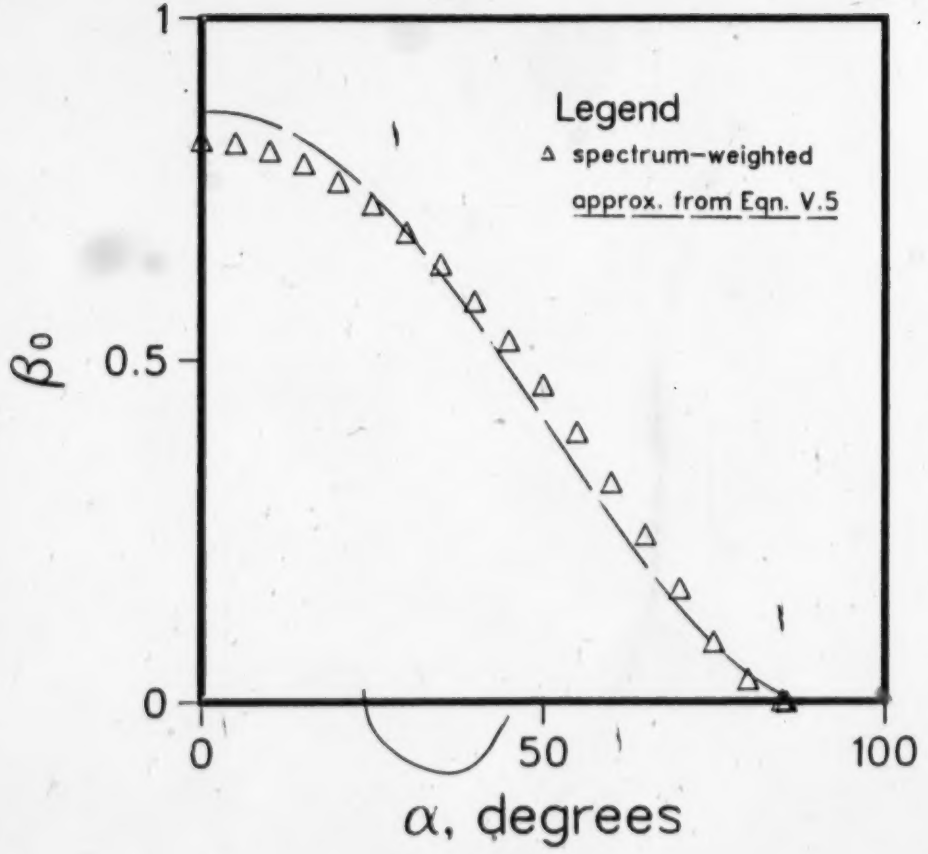
than actually occurs. For airfoils, the effect becomes important for $\beta_0$ less than about 0.8. For these cases, an alternative parameterization is proposed. It is derived from a polynomial fit to an average of several spectrum-weighted $\beta(\alpha)$ curves calculated by the author for droplet spectra observed in the FROST tunnel at the University of Alberta. The form of the parameterization is:

$$\beta(\alpha/\alpha_m) = \beta_0 \, (1 + 0.039(\alpha/\alpha_m) - 1.842(\alpha/\alpha_m)^2 - 0.543(\alpha/\alpha_m)^3 + 1.792(\alpha/\alpha_m)^4 - 0.444(\alpha/\alpha_m)^5, \qquad \text{V.5}$$

where $\beta_0 = \beta_{0,mvd}$, and

$$\alpha_m = \alpha_{max,mvd} + 0.13 \exp(-mvd \,/\, 70). \qquad \text{V.6}$$

Here $\alpha_m$ is again in radians, and mvd is in $\mu$m. In Figures V.5 and V.6, the approximation of Equation V.5 is compared with spectrum-weighted distributions for two other observed droplet spectra, which appear in Table A.1 as "Battan and Reitan stratus" (Battan and Reitan, 1957), and "FROST 3", a spectrum measured in the FROST tunnel. These figures also show the difference in $\beta_0$ between the spectrum-weighted value and that derived from the mvd for these cases.



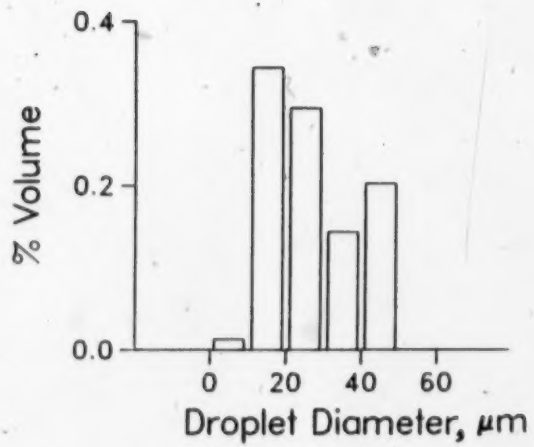




Figure V.5 - Comparison of spectrum-weighted collision efficiencies and Equation V.5 for the spectrum shown (mvd = 26.2 μm) on a NACA 0015, U = 50 m/sec, chord = 0.2 m.

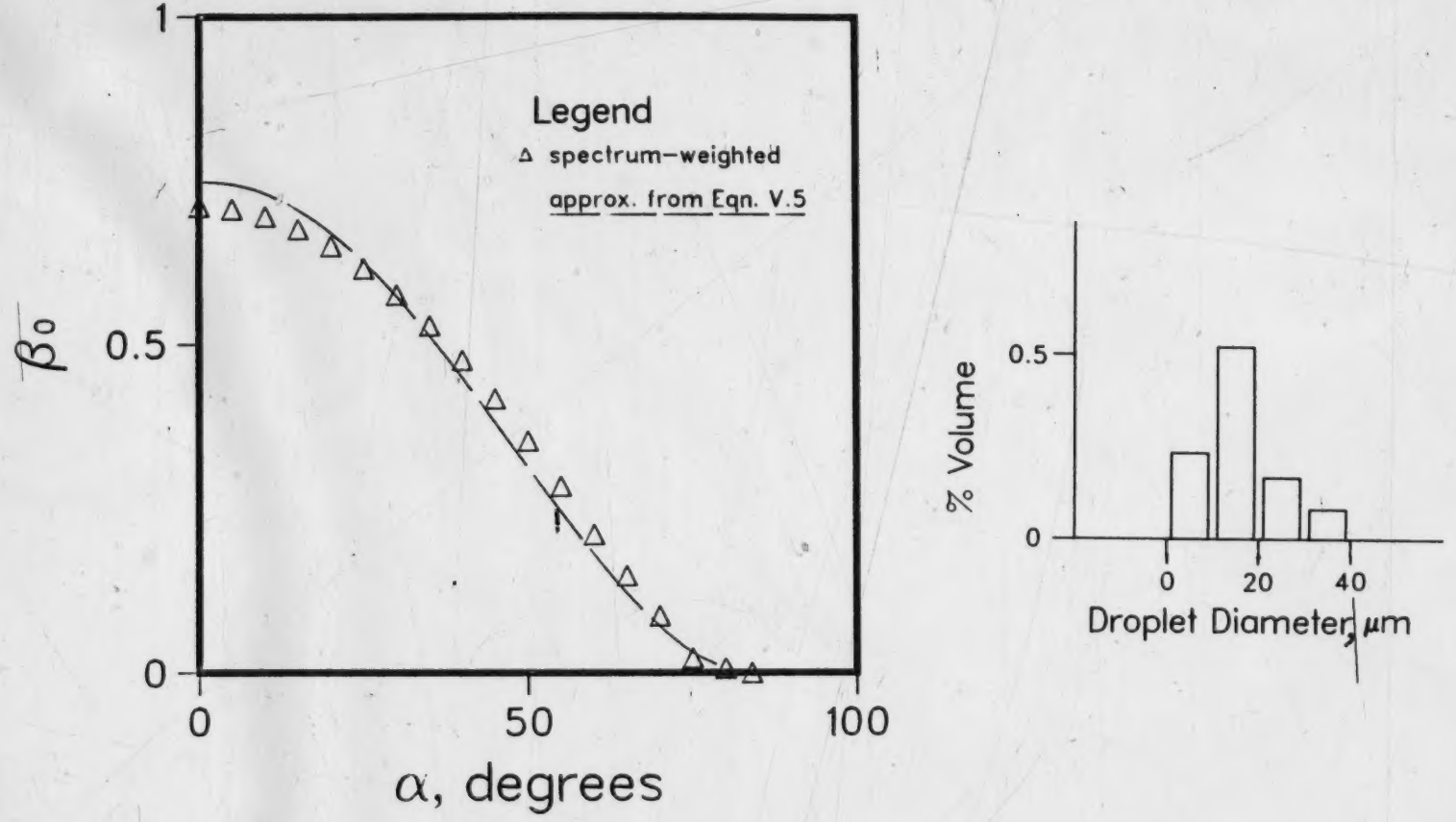



Figure V.6 - Comparison of spectrum-weighted collision efficiencies and Equation V.5 for the spectrum shown (mvd = 17.2 μm) on a NACA 0015, U = 50 m/sec, chord = 0.2 m.

When the angle of attack ($\Phi$) is non-zero, Oleskiw's numerical results have been fitted with the following function:

$$\beta(\alpha') = 0.95\ \beta_0 \cos(\alpha'), \qquad V.7$$

where $\beta_0$ is calculated as for the $\Phi = 0$ case, and

$$\alpha' = ((\pi/2)\ (\alpha_0 - \alpha)\ /\ (\pi/2 - \alpha)). \qquad V.8$$

The position of maximum collision efficiency has been designated by $\alpha_0$, and may be approximated by:

$$\alpha_0 = \pi \log_{10} ((180\ \Phi/\pi) + 1)\ /\ (0.180 \exp(30.5\ h_0)), \qquad V.9$$

where $\alpha_0$ and $\Phi$ are in radians.

Unfortunately, there are only two numerical cases currently available (from Oleskiw, 1982) to test this fit, as illustrated in Figures V.7 and V.8. Further calculations at some time in the future may help to refine the approximation. In the meantime, it does give a rough estimate for the $\beta(\alpha)$ distribution when $\Phi$ is non-zero, which is otherwise very costly to compute.

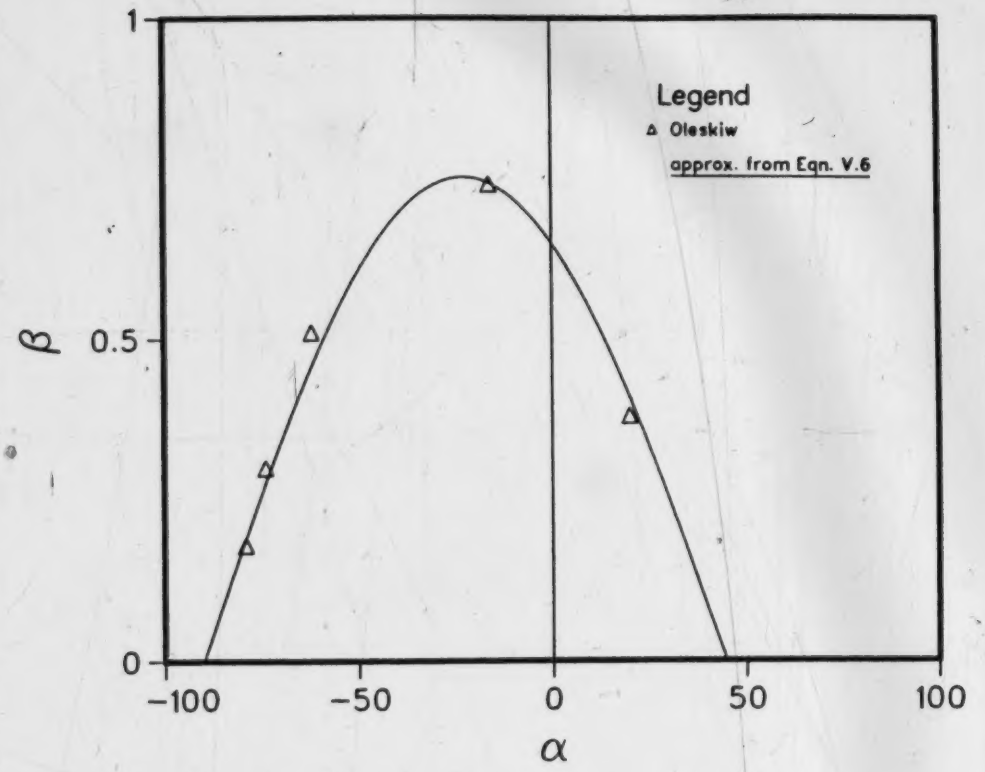



Figure V.7 - Comparison of Oleskiw's numerical results and Equation V.7, for collision efficiency on a NACA 0015 at an angle of attack = 8°, U = 30.5 m/sec, Chord = 0.169 m, $D_d$ = 20 μm.

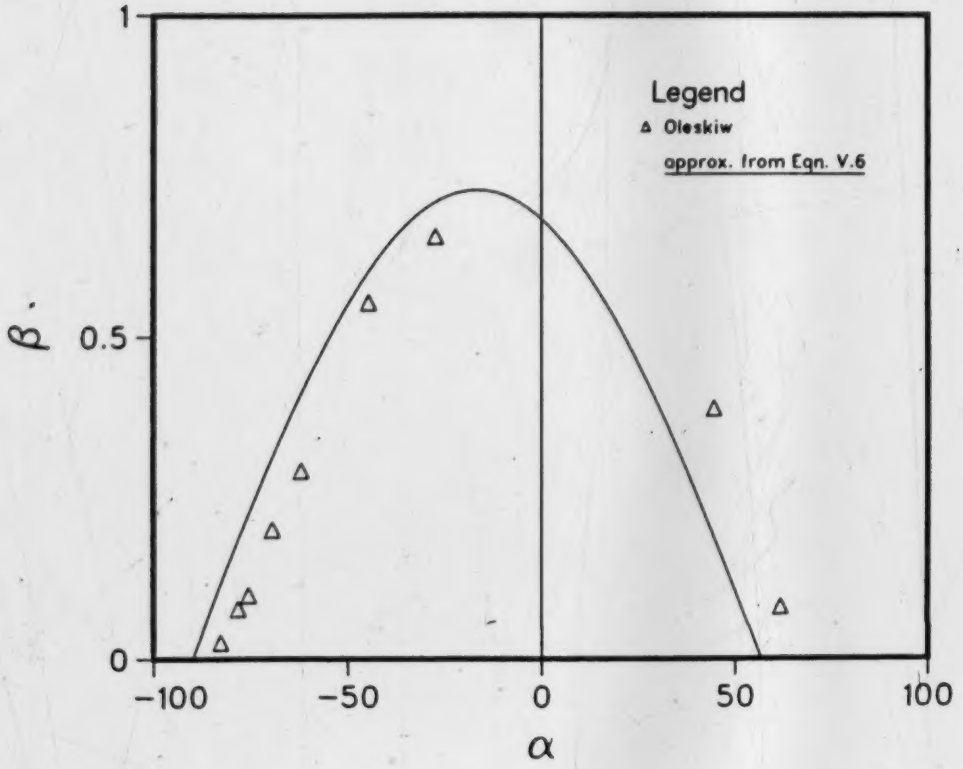



Figure V.8 - Comparison of Oleskiw's numerical results and Equation V.7, for collision efficiency on a Joukowski 15% at an angle of attack = 4°, U = 78.2 m/sec, Chord = 0.33 m, $D_d$ = 18.6 μm.

**E. Integration of Droplet Trajectories**

The equations of motion of a cloud droplet carried by the air stream, in a more general form than for the cylinder problem (Equation II.6), are:

$$dv_x/dt = -3/4\ (\rho_a/\rho_d)\ (C_d/D_d)\ V\ (v_x - u_x)$$

$$dv_y/dt = -3/4\ (\rho_a/\rho_d)\ (C_d/D_d)\ V\ (v_y - u_y) \qquad \text{V.10}$$

In this steady-state formulation the additional term due to gravity has been neglected, as has the "history" term, arising from the droplet's acceleration with respect to the flow. Neglect of these terms is justified by the results of Oleskiw (1982), and also by examination of the "acceleration modulus", (Norment, 1980):

$$N_A = D_d\ \left|\frac{dv}{dt}\right|\ /\ v^2 \qquad \text{V.11}$$

If this quantity is less than about 0.01, the history term may be safely neglected. The results of Oleskiw and of Norment showed that this limit is not exceeded in most aeronautical modelling situations.

The drag coefficient is again estimated from the the expressions of Beard and Pruppacher (1969), Equation II.9. The droplet trajectories are integrated by the Heun method (Mesinger and Arakawa, 1976), similar to the calculations in

Chapter II, except for the determination of the air speeds $u_x$ and $u_y$.

The streamfunction $\psi$ of the potential flow field about the current accretion shape is numerically calculated following the method of Kennedy and Marsden (1976). This is a version of the surface singularity method originated by Hess and Smith (1967), in which the airfoil surface is represented by a set of point vortices. Kennedy and Marsden have slightly altered, at the trailing edge, the usual boundary condition of a constant streamfunction along the airfoil surface. They claim this gives a very accurate result with a minimum of computing time. Full details are available in their paper. The air speeds may then be approximated from a finite-difference calculation of the streamfunction derivatives, with the spatial differences equal to the droplet diameter.

The droplet's initial x-coordinate is five chord lengths upstream of the nose, and the initial separation of the two trajectories, $\Delta y$, is $10^{-8}$ metres. An optimum time step was found to be $\Delta t = 0.0025$ (U/chord). The justification for these choices is given by Table V.1, which examines the sensitivity of the model results to changes in these parameters.

As the droplet approaches the airfoil, the model begins testing for its position with respect to the current accretion surface. At each time step $\Delta t$, the array of accreting surface points is searched for the pair of

| $\Delta t$ (chord/U) | $\beta_0$ ($x_0$=10• chord) | $\beta_0$ ($x_0$= 5 • chord) |
|---|---|---|
| 0.005 | 0.787 | 0.812 |
| 0.0033 | 0.794 | 0.804 |
| 0.0025 | 0.796 | 0.801 |
| 0.002 | 0.798 | 0.799 |

Table V.1 - Sensitivity of the integration results to the time step size $\Delta t$, and the initial x-position $x_0$. Trajectory is for a 20 micron drop impinging on a NACA 0015 airfoil of 0.213 m chord, in a 61 m/sec airstream.

adjacent points which bracket the current y-coordinate of the droplet. The droplet's x-coordinate is then compared to the x-coordinates of these two points. If either or both of them are greater than the droplet's x-coordinate, the current droplet position lies inside the accretion surface.

The collision point is then estimated by solving for the intersection of the straight line segments, one joining the two bracketing surface points, and one joining the current and immediately previous droplet positions (see Figure V.9). The arc length $\Delta\ell$ separating the two collision points on the surface is also approximated as a straight line. $\beta_0$ is then just $\Delta y/\Delta\ell$.

All of these straight line approximations, and the simple integration method, are sufficiently accurate because of the very small distances between surface points near the nose, the small $\Delta t$ and initial $\Delta y$ of the trajectories.

A comparison of the integration results with those of Oleskiw for his Case 55 shows very good agreement. Oleskiw calculates $\beta_0 = 0.805$ for a 20 $\mu$m drop impinging on a 0.213 m chord NACA 0015, at 0° angle of attack and 61 m/sec. The present model result is 0.801 for the same initial conditions. The small difference is easily accounted for by the use of a slightly different formulation for drag coefficient and by Oleskiw's method of approximating $\beta_0$. This is from a spline fit to several $\beta(\alpha)$ points that are distributed along the surface, away from the stagnation line.

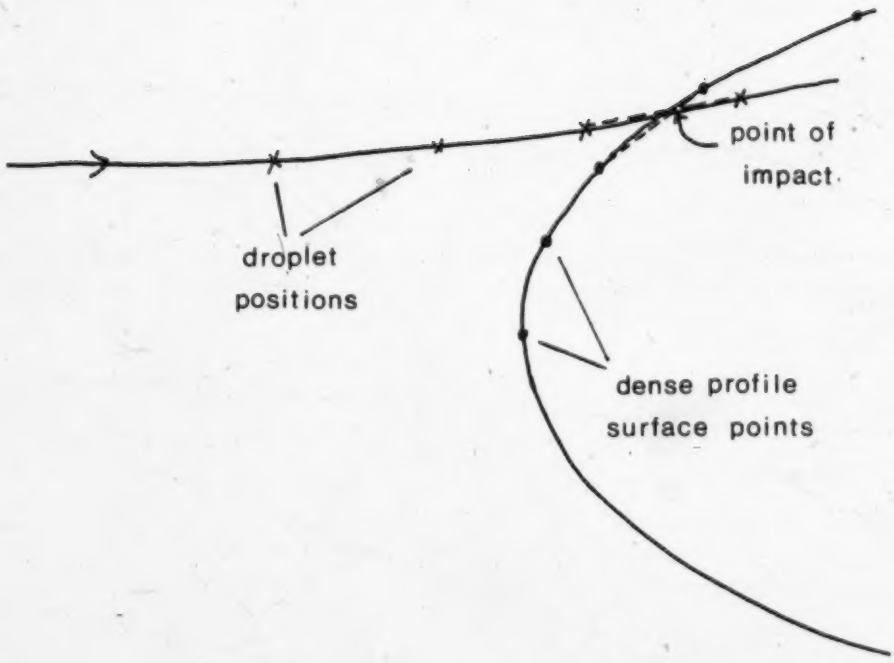



Figure V.9 - Determination of droplet impact point on the surface.

**F. Local Density Variations**

Once $\beta_0$ has been determined, $\alpha_{max}$ and the distribution of local collision efficiency may be approximated from Equations V.3, V.5 or V.7. The local ice thickness (Th) accreted over a layer time step $\tau$ at each surface point is then given by the dry growth equation:

$$Th = \beta(\alpha)\ U\ lwc\ \tau\ /\ \rho(\alpha), \qquad V.12$$

where now a suitable distribution for $\rho(\alpha)$ must be found.

This would ideally be done using the same techniques that were used in Chapter IV, to investigate circular cylinder accretion densities by direct measurement of local densities and of layer thicknesses. However, the wind tunnel experiments on airfoils which were performed in the FROST tunnel for this study, were grown on airfoils with thicknesses less than one inch. This is too small to allow effective measurements by the hole drilling technique, and the layers outlined by paint spray for some of the experiments were too thick to allow good results using the layer method.

Therefore, an approximate distribution was derived by plugging the above-mentioned circular cylinder curve into the model, and adjusting it until it could produce reasonable profiles. The result works best when used with the 'simulated spectrum' parameterization for local

collision efficiency. We emphasize that this distribution is not yet supported by any empirical evidence, and that it should be adjusted further when such evidence is available. For the present however, a good estimate for the density distribution is:

$$\rho(\alpha/\alpha_m) = \rho_0 \, (1 - 0.163 \, (\alpha/\alpha_m) + 0.235(\alpha/\alpha_m)^2 - 2.357(\alpha/\alpha_m)^3 + 1.374(\alpha/\alpha_m)^4 + 0.019(\alpha/\alpha_m)^5), \qquad \text{V.13}$$

where $\rho_0$ is assumed, for simplicity, to be always 890 $kg/m^3$. In certain conditions, the maximum density may be less than this value, but the model has no capability to determine this, at present.

The mass and mean density of each layer are calculated in the same way as for the circular cylinder described in Chapter III.

## G. Model Results

Available data for model validation include the numerical results of Oleskiw, and a number of wind tunnel experiments carried out in the University of Alberta FROST tunnel, at NASA/Lewis and described in Olsekiw (1982b), at the National Aeronautical Establishment by Stallabrass (1958), and at the National Research Council Low Temperature Laboratory by Lozowski and Stallabrass (1978). In all of the experimental cases, the droplet sizes were determined by the

oil slide method, and have been corrected by us according to the calibration of Makkonen and Stallabrass (1984) discussed in Appendix III. An exception is the case taken from Stallabrass (1958), for which the liquid water content given by Stallabrass was derived from the observed ice thickness and the oil slide measurement of mvd. Stallabrass' numbers have thus been retained for both the droplet size and lwc. In all other cases, however, the accretion conditions were measured independently of the resulting thicknesses.

Figure V.10 illustrates the modelled profiles of Oleskiw and of the present model for identical accretion conditions. Since a constant local density was used, the agreement in shape of the layers between the two models reflects the accuracy of the parameterized collision efficiency distribution, for this case Equation V.3. The present model also lacks the surface instabilities which are already appearing in the Oleskiw model profile, arising from the spline curve approximation to the layer surface. Most importantly, the bottom profile of Figure V.10 represents a reduction in computational costs of a factor of ten over the Oleskiw model.

The conditions of Figure V.10 were chosen to agree with those of an experimental case grown in the NASA/Lewis tunnel. This experimental profile appears in Figure V.11, together with a corresponding model simulation using a constant local density, and one using the 'simulated spectrum' version of the local collision efficiency

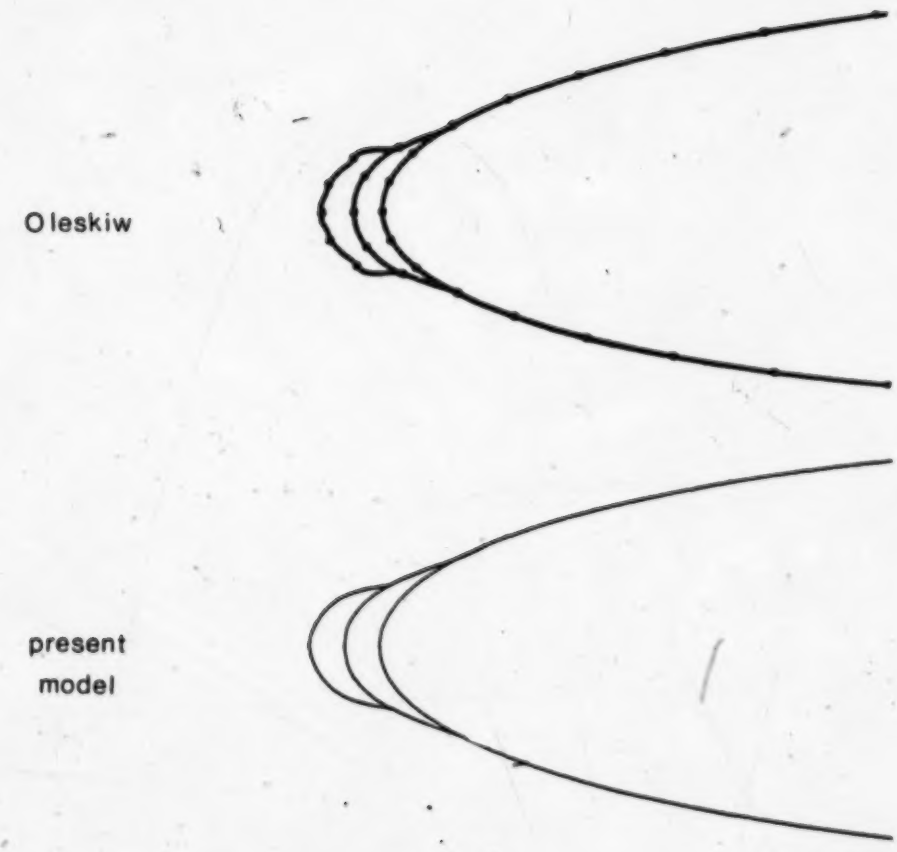



Figure V.10 - Comparison of present model results with those of Oleskiw (1982b) for a constant density accretion on a NACA 0015 at $\Phi = 0°$, U = 58 m/sec, lwc = 1.02 g/m$^3$, chord = 0.533 m, t = 5 min. A monodisperse droplet population at 12 μm diameter is assumed, and an air temperature of -15° C.

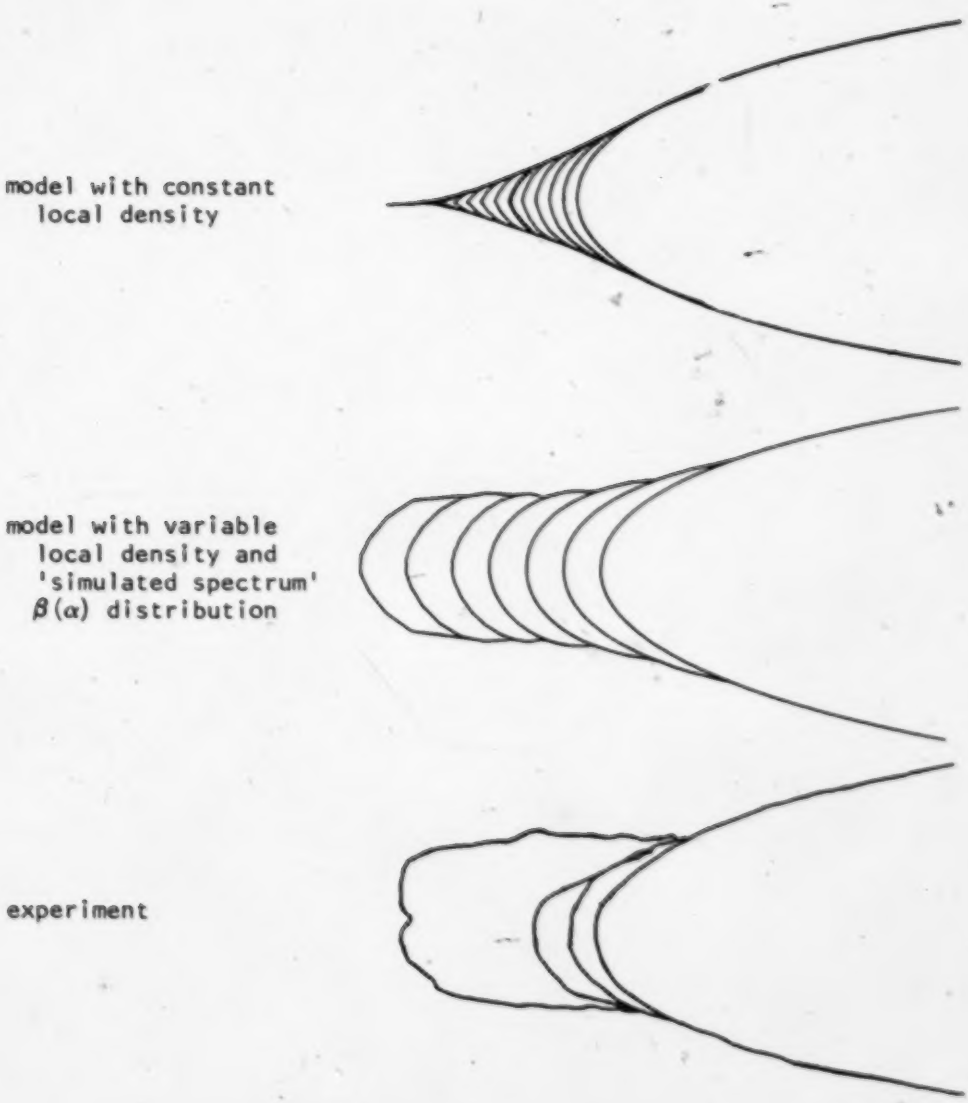



Figure V.11 - Comparison of present model results with an experimental accretion grown in the NASA/Lewis tunnel and described in Oleskiw (1982b), for the same conditions as in Figure V.10, but for a NACA 0012 airfoil, t = 15 min. Intermediate profiles at t = 2 and 5 minutes are also marked.

distribution, Equation V.5, together with the local density distribution of Equation V.12.

The two modelled profiles demonstrate a great improvement, in terms of a realistic accretion shape, in using the new local density and 'simulated spectrum' distributions over the old constant density and monodisperse spectrum versions. The model profiles are also seen to be free of surface instabilities and thus capable of accreting many more layers than the Oleskiw model.

Except for the forward surface of the experimental accretion, which seems to contain a small central trough, the agreement in shape is very good. Central troughs and ridges are features which are often seen to form on wind-tunnel accretions. They most probably arise from disturbances in the laminar flow pattern near the nose, as the ice layer alters its shape. The simple, potential flow model does not include such effects, and such features are not expected to show up in these simulations.

The experimental results of Lozowski and Stallabrass (1978) and of Stallabrass (1958) allow a test of Equation V.7, for local collision efficiency at non-zero angles of attack. Figure V.12 shows four experimental accretions and the corresponding model profiles, all including density variation and local collision efficiencies estimated for the $\Phi \neq 0$ case from Equation V.7. Since the method of approximating accretion limits for $\Phi \neq 0$ is somewhat crude,

Figure V.12 - Comparison of present model results with experimental profiles from Stallabrass (1958, top profile, NACA 0012) and Stallabrass and Lozowski (1978, lower three profiles, Air temperature is -15° C.

comparison in this region shows only rough agreement. However, the general agreement in shape and stagnation line thicknesses is encouraging. These cases at least illustrate that the approach taken in Equation V.9, to approximate the position of maximum collision efficiency $\alpha_0$, is worth pursuing and refining, especially given the large computational costs of determining this position from a series of droplet trajectory integrations.

For all of the cases mentioned so far, accretion masses have not been available. However, a series of wind-tunnel experiments were carried out in the University of Alberta FROST tunnel, for which was measured the mass of ice accreted on a 15 cm length of a NACA 0012 airfoil under various conditions. These results, with intermediate layers as marked by paint spray, are shown in Figure V.13 together with the modelled masses and profiles.

The differences between the model masses and the experimental masses for the two cases are 11 and 16 percent. This is of the same order as typical measurement errors for input parameters such as droplet size and liquid water content.

These examples have shown the usefulness of a simple approach toward numerical methods for ice accretion modelling. The present results are comparable to and much less costly than those of the Oleskiw model, and are of

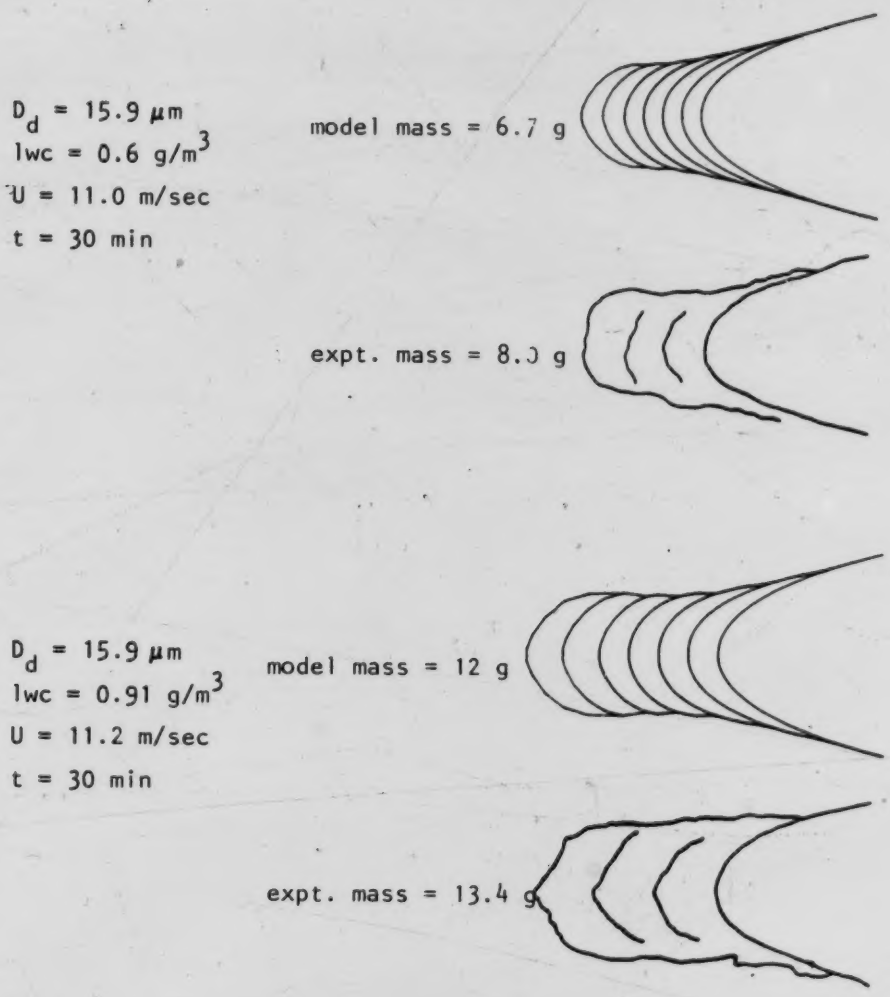

$D_d$ = 15.9 μm
lwc = 0.6 g/m$^3$
U = 11.0 m/sec
t = 30 min
model mass = 6.7 g
expt. mass = 8.0 g
$D_d$ = 15.9 μm
lwc = 0.91 g/m$^3$
U = 11.2 m/sec
t = 30 min
model mass = 12 g
expt. mass = 13.4 g

Figure V.13 - Comparison of present model results with experimental profiles and masses of accretions grown in The University of Alberta FROST tunnel on a NACA 0012 at $\Phi = 0°$, chord = 0.176 m, air temperature = -10° C.

sufficient accuracy to allow practical and realistic comparisons with rime experiments. Such methods will also help to make the modelling problem for wet icing more tractable.

## References


Abbott,I.H. and von Doenhoff, A.E. 1959: Theory of Wing Sections Including a Summary of Airfoil Data. Dover, 693 pp.

Beard, K.V. and Pruppach, H.R. 1969: A Determination of the Terminal Velocity and Drag of Small Water Drops by Means of a Wind Tunnel. *J. Atmos. Sci.*, 26, 1066-1072.

Cansdale, J.T. and Gent, R.W. 1983: Ice Accretion on Airfoils in Two-Dimensional Compressible Flow - A Theoretical Model. British Royal Aircraft Establishment Technical Report No. 82128, 31 pp. plus 33 fig.s.

Hess, J.L. and Smith, A.M.O. 1967: Calculation of Potential Flow about Arbitrary Bodies. *Progress In Aeronautical Sciences, Vol. 8.* Pergamon Press, 1-138.

Kennedy, J.L. and Marsden, D.J. 1976: Potential Flow Velocity Distributions on Multi-Component Airfoil Sections. *Can. Aero. and Space Jour.*, 22, 243-256.

MacArthur, C.D., Keller, J.L. and Luers, J.K. 1982: Mathematical Modelling of Ice Accretion on Airfoils. AIAA Publication No. 82-0284.

Makkonen, L. and Stallabrass, J.R. 1984: Ice Accretion on Cylinders and Wires. National Research Council, Division of Mechanical Engineering, Technical Report TR-LT-005, 50 pp.

Mesinger, F. and Arakawa, A. 1976: Numerical Methods used in Atmospheric Models, Vol. 1. W.M.O.GARP Publications Series No. 17, 64 pp.

Norment, H.G. 1980: Calculation of Water Drop Trajectories to and About Arbitrary Three-Dimensional Bodies in Potential Airflow. NASA Contractor Report 3291, 82 pp.

Oleskiw, M.M. 1982: Numerical Simulation of Icing on Airfoils. Ph.D. Thesis, The University of Alberta, 302 pp.

Oleskiw, M.M. 1982b: Contribution to NASA/Lewis Icing Workshop, unpublished.

Stallabrass, J.R. 1958: Icing Flight Trials of a Sikorsky H04S-2 Helicopter. National Aeronautical

Establishment, Laboratory Report LR-219, 25 pp.

Stallabrass, J.R. and Lozowski, E.P. 1978: Ice Shapes on Cylinders and Rotor Blades. Presented to NATO Panel X Symposium on Helicopter Icing, London.

## VI. Concluding Remarks

In the four separate, but interrelated studies of this thesis several improvements have been put forward for physical models of ice accretion. Since conclusions were given within each chapter, these remarks will summarise the results, and give suggestions for future research.

Chapter II presented a re-calculation of the long-established work of Langmuir and Blodgett on collision efficiencies and other impingement parameters of circular cylinders. High-speed digital computer results for droplet trajectory integrations have provided a small improvement over the old analog computer results, but a larger improvement has been made in the application of a non-linear regression fit to the tabulated numerical values of collision efficiency, impact speed and accretion limit. This new analytical approximation could be very useful to modellers, who presently use a variety of functions to approximate the same thing, none of which is as accurate as the new expression.

These results are equally applicable to all types of icing, wet or dry, from transmission lines to marine structures, as long as it occurs on circular cylinders. A possibility for future work here is to extend the re-calculations to the spheres and flat ribbons also considered by Langmuir and Blodgett.

The circular cylinder rime model of Chapter III makes use of these new collision efficiencies, as well as of new

parameterizations for the distribution of local collision efficiency under various conditions. The efficiency and accuracy of these methods was well demonstrated in that chapter, by model results which reproduced realistic profiles, and accreted masses to within 10 percent of observation. The speed of the model; a few seconds on the University of Alberta Amdahl 580, and about 90 seconds on an IBM PC; make it highly practical.

The cylinder model also represents an improvement on older models in its version of local ice density distribution, derived in Chapter IV. Based on empirical data, this distribution is more realistic than previous versions, based on rotating cylinder correlations, have been. Since an attempt to validate the density distribution by direct measurement was inconclusive, further work, perhaps using different measurement techniques, would be valuable.

A logical extensions to the circular cylinder model would be the addition of gravitational and aerodynamic torque calculations in order to determine the correct rate of rotation. Another would be the determination of stagnation line density, following the method described in Chapter IV. The stagnation line density is required as a scaling parameter of the empirical density distribution. This would require a simple heat balance calculation at the stagnation line in order to determine the surface temperature there.

The airfoil model of Chapter V shows that parameterization methods similar to those of Chapter III may also be found for more difficult shapes, where the potential saving in computational costs is much larger. Here the parameterization for $\rho(\alpha)$ is in particular need of empirical validation, and the collision efficiency approximations would benefit from further verification and refinement using numerical results for a wider range of conditions.

The airfoil model has the capability to update the potential flow approximation, from which the $\beta_0$ parameter is derived, at intervals throughout the accretion period. Although a suggestion was made for the length of this interval, it was only an estimate. More numerical experiments should be carried out to discover the optimum interval, since this procedure is an expensive one.

Both models fail in the prediction of the troughs and ridges which sometimes form along the stagnation line of experimental accretions, often after the formation of a flat forward surface. An interesting exercise would be to investigate the flow patterns and droplet trajectories about these flat surfaces, in an attempt to understand how such features are formed. Experimental data should be gathered on the conditions under which troughs and ridges form, since they are not always present. Such studies may discover ways to model their formation.

# Appendix I

## Program Listing: GRAZE

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
1.    C ********************** GRAZE ************************
2.    C
2.2   C     Karen J. Finstad      June, 1986
2.4   C
3.    C     This program calculates trajectory pairs for water
4.    C  droplets in an air flow field around a circular cylinder,
5.    C  and determines the stagnation line collision efficiency,
6.    C  maximum accretion angle and overall collision efficiency.
7.    C  The air is assumed to be at a temperature of -10 degrees
8.    C  C, and a pressure of 100 kPa.
9.    C
10.   C     The equation of motion of the droplet is integrated
11.   C  using the Heun scheme (Mesinger, F. and Arakawa, A. 1976.
12.   C  Numerical Methods Used in Atmospheric Models, Vol. 1.
13.   C  W.M.O. GARP Publications Series No. 17, 64pp.) The time
14.   C  step is set to 1/400 of the time to travel one cylinder
15.   C  radius at the free stream speed. A maximum of 7000 steps
16.   C  are allowed, by which time the droplet is assumed to have
17.   C  collided with the cylinder or to have passed it.
18.   C
19.   C     Note that the droplet's initial x-coordinate is set
20.   C  to be 10 cylinder radii upstream of the cylinder axis.
21.   C  The initial x-component speed is set equal to the air's
22.   C  x-component, and the y-component is set to half the
23.   C  air's y-component.
24.   C
25.   C     A first trajectory is integrated beginning at the
26.   C  input initial y-coordinate, which is then incremented
27.   C  by a very small amount (DELTAY) for the second trajectory.
28.   C  If both trajectories intercept the cylinder, the normal
29.   C  impact velocity, impact angle, impact position, and local
30.   C  collision efficiency are calculated.
31.   C
32.   C     The grazing trajectory is determined by successively halving
33.   C  the distance between a pair of initial y-coordinates which
34.   C  bracket the initial y-coordinate of the grazing trajectory.
35.   C  The initial values of y-min and y-max are input by the user;
36.   C  a judicious choice will decrease the number of trajectories
37.   C  needed to determine the grazing one.
38.   C     If a Stokes flow calculation is requested (i.e. Langmuir phi
39.   C  parameter = 0), the drag coefficient is set to 24 / Reynolds
40.   C  number. Otherwise, the formulation of Beard and Pruppacher is
41.   C  used for the drag coefficient.
42.   C
43.   C     All operations are carried out in double precision.
44.   C  Output is written to *MSOURCE.
45.   C
46.   C     To compile on MTS, enter:
47.   C     $RUN *FORTRANVS SCARDS=GRAZE SPUNCH=GRAZERUN
48.   C
49.   C     To run, enter $RUN GRAZERUN, You will be prompted for
50.   C  the following input parameters:
51.   C
```

```
      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
C     Droplet DIAMETER in microns
C
C     Cylinder RADIUS in metres
C
C     Free stream air speed in metres per second
C
C     Initial droplet y position in metres
C
C     Initial value for y-min
C
C     Initial value for y-max
C
C     y(es) or n(o) for Stokes flow option
C
C  ******************** Subprograms **********************
C
C
C     SUBROUTINE TRAJEC - calculates droplet trajectories
C       by integrating the droplet equations of motion by
C       the trapezoid rule with one iteration step.
C
C     SUBROUTINE ITERAT- calculates the air speeds (from
C       the analytical flow solution) and droplet
C       accelerations (from the equations of motion) for
C       the current iteration and time step.
C
C     SUBROUTINE COLLID -calculates final impact position
C       if the droplet collides with the surface.
C
C
C  ***************** Variable Dictionary ******************
C
C     ARC1,      - arc lengths along cylinder surface
C     ARC2         from stagnation line to the impact points
C                  of the two trajectories
C
C     BETA       - local collision efficiency
C
C     CYLRAD     - cylinder radius in metres (INPUT)
C
C     DELTAT     - time step in seconds
C
C     DELTAY     - difference in initial y coordinate of
C                  the two trajectories, in metres
C
C     DROPDI     - droplet diameter, input in microns, but
C                  converted to metres (INPUT)
C
C     FINVX,     - average values of the final velocities
C     FINVY
C
C     FINVX1,    - x-components of droplet velocity at
C     FINVX2       impact
C
C     FINVY1,    - as above, for y-components
C     FINVY2
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                107.    C
                108.    C     HALFPI     - pi / 2 (FORTRAN77 constant parameter
                109.    C
                110.    C     HIT1,      - logical flags for each trajectory
                111.    C     HIT2         which indicate whether a collision
                112.    C                  with the cylinder has occurred
                112.1   C
                113.    C     NOTRAJ     - no. of trajectories used to find grazing traj.
                114.    C
                115.    C     NSTEPS     - maximum number of time steps to be
                116.    C                  integrated
                116.1   C
                117.    C     PHI        - angle of trajectory to the surface
                118.    C                  normal at impact
                118.1   C
                119.    C     STK        - input character variable to set Stokes
                120.    C                  flow flag
                120.08  C
                120.5   C     STOKES     - logical flag for Stokes flow option
                120.7   C
                120.71  C     THETA      - average of THETA1 and THETA2
                120.8   C
                120.81  C     THETA1,    - position angles of the impact points
                120.87  C     THETA2       of the two trajectories
                120.93  C
                121.    C     VEL        - free stream air speed in metres/sec (INPUT)
                122.    C
                122.11  C     VNORM      - droplet velocity normal to the cylinder
                122.4   C                  surface at impact
                122.7   C
                123.    C     XINIT      - initial droplet x-coordinate in metres
                124.    C
                125.    C     YINIT      - initial droplet y-coordinate in metres (INPUT)
                126.    C
                144.    C     YMIN,      - minimum and maximum initial y-coordinates
                145.    C     YMAX         which bracket the graze coordinate
                146.    C
                147.    C ***********************************************************
                148.    C
ISN        1    149.            DOUBLE PRECISION DROPDI,VEL,CYLRAD,XINIT,YINIT,FINVX,
                150.           +                 FINVY,VNORM,PHI,THETA,THETA1,THETA2,
                151.           +                 BETA,FINVX1,FINVX2,FINVY1,FINVY2,
                152.           +                 ARC1,ARC2,DELTAT,DELTAY,YMIN,YMAX
                153.    C
ISN        2    154.              INTEGER        NSTEPS,NOTRAJ
                155.    C
ISN        3    156.              LOGICAL        HIT1,HIT2,STOKES
                157.    C
ISN        4    158.              CHARACTER*1    RESPNS,STK
                159.    C
ISN        5    160.              PARAMETER (HALFPI = 3.1415926535897932/2.0)
                161.    C
                162.    C  Prompt user for input parameters
                163.    C
ISN        6    164.         5    WRITE (6,10)
ISN        7    165.        10    FORMAT (///,' ********* Droplet Trajectories about a',
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                166.          + ' Cylinder *********',//)
                167.    C
ISN       8     168.          WRITE (6,*) 'Droplet DIAMETER in microns?'
ISN       9     169.          READ  (5,*) DROPDI
ISN      10     170.          DROPDI = DROPDI * 1.0E-6
                171.    C
ISN      11     172.          WRITE (6,*)
ISN      12     173.          WRITE (6,*) 'Cylinder RADIUS in metres?'
ISN      13     174.          READ  (5,*) CYLRAD
                175.    C
ISN      14     176.          WRITE (6,*)
ISN      15     177.          WRITE (6,*) 'Free stream air speed in m/sec?'
ISN      16     178.          READ  (5,*) VEL
                179.    C
ISN      17     180.          WRITE (6,*)
ISN      18     181.          WRITE (6,*) 'Initial droplet y-coordinate in metres?'
ISN      19     182.          READ  (5,*) YINIT
                183.    C
ISN      20     184.          WRITE (6,*)
ISN      21     185.          WRITE (6,*) 'Initial value of ymin?'
ISN      22     186.          READ(5,*) YMIN
                187.    C
ISN      23     188.          WRITE (6,*)
ISN      24     189.          WRITE (6,*) 'Initial value of ymax?'
ISN      25     190.          READ (5,*) YMAX
                191.    C
ISN      26     192.          WRITE (6,*)
ISN      27     193.          WRITE (6,*) 'Do you wish a Stokes flow calculation? y/n'
ISN      28     194.          READ  (5,15) STK
ISN      29     195.       15 FORMAT (A1)
                196.    C
                197.    C  Set flag for Stokes flow case
                198.    C
ISN      30     199.          IF ((STK .EQ. 'y') .OR. (STK .EQ. 'Y')) THEN
ISN      31     200.             STOKES = .TRUE.
ISN      32     201.          ELSE
ISN      33     202.             STOKES = .FALSE.
ISN      34     203.          ENDIF
                204.    C
                205.    C  Initialize the total number of trajectory pairs.
                206.    C
ISN      35     207.          NOTRAJ = 0
ISN      36     208.       100 CONTINUE
                209.    C
                210.    C  Set maximum number of time steps , initial x-coordinate, and
                211.    C  the difference in initial y-coordinates. Increment number
                212.    C  of trajectories.
                213.    C
ISN      37     214.          NOTRAJ = NOTRAJ + 1
ISN      38     215.          NSTEPS = 7000
ISN      39     216.          DELTAY = 1.0D-8
ISN      40     217.          XINIT  = 10 * CYLRAD
                218.    C
                219.    C  Calculate time step
                220.    C
ISN      41     221.          DELTAT = CYLRAD / (VEL * 200)
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                222.    C
                223.    C  Integrate first trajectory
                224.    C
ISN       42    225.          CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CYLRAD,DELTAT,NSTEPS,
                226.         +             STOKES,THETA1,ARC1,HIT1,FINVX1,FINVY1)
                227.    C
                228.    C  Correct initial y position and integrate second trajectory
                229.    C
ISN       43    230.          YINIT = YINIT + DELTAY
ISN       44    231.          CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CYLRAD,DELTAT,NSTEPS,
                232.         +             STOKES,THETA2,ARC2,HIT2,FINVX2,FINVY2)
                233.    C
                234.    C  Test for result of trajectory (miss, graze or collision)
                235.    C
ISN       45    236.          IF ((HIT1) .AND. (HIT2)) THEN
                237.    C
                238.    C       Calculate impact parameters and local collision efficiency
                239.    C
ISN       46    240.            IF (YINIT-DELTAY .NE. 0.0) YMIN = YINIT - DELTAY
ISN       48    241.            BETA   = DELTAY / (ARC2 - ARC1)
ISN       49    242.            THETA = (THETA1 + THETA2) / 2.0
ISN       50    243.            FINVX = (FINVX1 + FINVX2) / 2.0
ISN       51    244.            FINVY = (FINVY1 + FINVY2) / 2.0
ISN       52    245.            PHI    = HALFPI - DATAN(FINVX/FINVY) + THETA
ISN       53    246.            VNORM = DSQRT((FINVX * FINVX) + (FINVY * FINVY))
                247.           +             * DCOS(PHI)
                248.    C
                249.    C  Write results to screen
                250.    C
ISN       54    251.            WRITE (6,20)THETA,VNORM,PHI,BETA
ISN       55    252.      20    FORMAT (//,' Both trajectories collided with cylinder.',
                253.         +          //,' Position angle at impact point: ',F7.3
                254.         +          //,' Velocity normal to surface at impact: ',F7.3,
                255.         +          //,' Angle of trajectory to surface normal: ',F7.3
                256.         +          //,' Local collision efficiency: ',F5.3,//)
ISN       56    257.          ELSE IF (HIT1) THEN
ISN       57    258.            WRITE (6,30) THETA1
ISN       58    259.      30    FORMAT (//,' First trajectory grazed cylinder.',
                260.         +          //,' Surface angle of max accretion point: ',F7.3,//)
ISN       59    261.          WRITE (6,*) ' overall coll. eff.:', (YINIT - DELTAY)/CYLRAD
ISN       60    262.          WRITE (6,*) ' no of trajectories: ',NOTRAJ
ISN       61    263.            GO TO 200
ISN       62    264.          ELSE
ISN       63    265.            YMAX = YINIT -DELTAY
ISN       64    266.            WRITE (6,40)
ISN       65    267.      40    FORMAT (//,' Both trajectories missed the cylinder.')
ISN       66    268.          ENDIF
                269.    C
                270.    C  Determine new initial position and start next trajectory pair
                271.    C
ISN       67    272.          YINIT = (YMAX + YMIN) / 2.0
ISN       68    273.          GO TO 100
                274.    C
ISN       69    275.      200 STOP
ISN       70    276.          END
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:13:40      NAME: MAIN    PAGE:    6
*STATISTICS*   SOURCE STATEMENTS = 69, PROGRAM SIZE = 2732 BYTES, PROGRAM NAME = MAIN    PAGE:    1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
              277.    C
              278.    C ****************************************************************
              279.    C *    SUBROUTINE TRAJEC                                         *
              280.    C *                                                              *
              281.    C *       Integrates trajectory of droplet whose initial         *
              282.    C *    position is specified in the parameter list. Calls        *
              283.    C *    ITERAT for iterative calculations, and COLLID to          *
              284.    C *    determine impact parameters for colliding trajectories.   *
              285.    C *                                                              *
              286.    C *    INPUT VARIABLES:                                          *
              287.    C *                 XINIT  - initial x position                  *
              288.    C *                 YINIT  - initial Y position                  *
              289.    C *                 DROPDI - droplet diameter                    *
              290.    C *                 CYLRAD - cylinder radius                     *
              291.    C *                 VEL    - free stream air speed               *
              292.    C *                 DELTAT - time step                           *
              293.    C *                 NSTEPS - maximum number of time steps        *
              294.    C *                 STOKES - logical flag for Stokes flow        *
              295.    C *                                                              *
              296.    C *    OUTPUT VARIABLES:                                         *
              297.    C *                 THETA  - surface angle of impact             *
              298.    C *                 ARC    - arc length along cylinder surface   *
              299.    C *                          from stagnation line to impact      *
              300.    C *                          point                               *
              301.    C *                 HIT    - logical flag set to .TRUE. if       *
              302.    C *                          a collision occurs                  *
              303.    C *                 FINVX  - droplet x velocity on impact        *
              304.    C *                 FINVY  - droplet y velocity on impact        *
              305.    C *                                                              *
              306.    C *    OTHER VARIABLES:                                          *
              307.    C *                 X,Y    - arrays containing x,y positions     *
              308.    C *                          for each time step                  *
              309.    C *                 VX,VY  - arrays containing x,y components    *
              310.    C *                          of droplet speed for each step      *
              311.    C *                 DVX,   - arrays containing x,y components    *
              312.    C *                 DVY      of droplet accelerations            *
              313.    C *                                                              *
              314.    C****************************************************************
              315.    C
ISN       1   316.          SUBROUTINE TRAJEC(XINIT,YINIT,DROPDI,VEL,CYLRAD,DELTAT,
              317.         +                  NSTEPS,STOKES,THETA,ARC,HIT,FINVX,FINVY)
              318.    C
ISN       2   319.          DOUBLE PRECISION X(7000),Y(7000),
              320.         +                 VX(7000),VY(7000),DVX(7000),DVY(7000)
              321.    C
ISN       3   322.          DOUBLE PRECISION XINIT,YINIT,DROPDI,CYLRAD,VEL,DELTAT,
              323.         +                 THETA,ARC,FINVX,FINVY
              324.    C
ISN       4   325.          INTEGER          NSTEPS
              326.    C
ISN       5   327.          LOGICAL          HIT,STOKES
              328.    C
              329.    C  Initialize values for integration
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                330.    C
ISN       6     331.          X(1)  = -XINIT
ISN       7     332.          Y(1)  = YINIT
ISN       8     333.          VY(1) = (DABS(2 * VEL * CYLRAD * CYLRAD * X(1) * Y(1)) /
                334.         +          (((X(1) * X(1)) + (Y(1) * Y(1)))**2)) / 2.0
ISN       9     335.          VX(1) = VEL
ISN      10     336.          I     = 1
ISN      11     337.          HIT   = .FALSE.
                338.    C
                339.    C  Begin integration
                340.    C
ISN      12     341.      100  CONTINUE
                342.    C
                343.    C  Calculate current step air speeds and accelerations
                344.    C
ISN      13     345.            CALL ITERAT(X(I),Y(I),VX(I),VY(I),VEL,CYLRAD,DROPDI,
                346.         +                STOKES,DVX(I),DVY(I))
                347.    C
                348.    C  Calculate next step droplet speeds and positions
                349.    C
ISN      14     350.            VX(I+1) = VX(I) - (DELTAT * DVX(I))
ISN      15     351.            VY(I+1) = VY(I) + (DELTAT * DVY(I))
ISN      16     352.            X(I+1)  = X(I) + (VX(I) * DELTAT) - (0.5 * DVX(I)
                353.         +              * (DELTAT * DELTAT))
ISN      17     354.            Y(I+1)  = Y(I) + (VY(I) * DELTAT) + (0.5 * DVY(I)
                355.         +              * (DELTAT * DELTAT))
                356.    C
                357.    C  Calculate next step air speeds and accelerations
                358.    C
ISN      18     359.            CALL ITERAT(X(I+1),Y(I+1),VX(I+1),VY(I+1),VEL,
                360.         +              CYLRAD,DROPDI,STOKES,DVX(I+1),DVY(I+1))
                361.    C
                362.    C  Iterate calculations of next step droplet speed and
                363.    C  position using averaged values of accelerations
                364.    C
ISN      19     365.            VX(I+1) = VX(I) - ((DELTAT / 2.0) * (DVX(I) +
                366.         +                DVX(I+1)))
ISN      20     367.            VY(I+1) = VY(I) + ((DELTAT / 2.0) * (DVY(I) +
                368.         +                DVY(I+1)))
ISN      21     369.            X(I+1) =  X(I) + (VX(I) * DELTAT) - (0.25 *
                370.         +                (DVX(I) + DVX(I+1)) * (DELTAT * DELTAT))
ISN      22     371.            Y(I+1) =  Y(I) + (VY(I) * DELTAT) + (0.25 *
                372.         +                (DVY(I) + DVY(I+1)) * (DELTAT * DELTAT))
                373.    C
                374.    C  Test whether trajectory has intercepted the cylinder
                375.    C
ISN      23     376.            IF (DSQRT((X(I+1)**2) + (Y(I+1)**2)) .LE. CYLRAD) THEN
ISN      24     377.               HIT   = .TRUE.
ISN      25     378.               FINVX = (VX(I) + VX(I+1)) / 2.0
ISN      26     379.               FINVY = (VY(I) + VY(I+1)) / 2.0
ISN      27     380.               CALL COLLID(X(I),X(I+1),Y(I),Y(I+1),CYLRAD,
                381.         +                 THETA,ARC)
ISN      28     382.            ELSE IF (I .LT. (NSTEPS - 1)) THEN
ISN      29     383.               I = I + 1
ISN      30     384.               GO TO 100
ISN      31     385.            ENDIF
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
               386.     C
ISN      32    387.           RETURN
ISN      33    388.           END

*STATISTICS*   SOURCE STATEMENTS = 33, PROGRAM SIZE = 338354 BYTES, PROGRAM NAME = TRAJEC    PAGE:    7.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 2 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                C ***************************************************************
                C *                                                             *
                C *  SUBROUTINE ITERAT  -  calculates for the current step:     *
                C *      air speed components from analytical solution to       *
                C *      flow around a cylinder, the droplet Reynolds           *
                C *      number, drag coefficient according to the formula-     *
                C *      tion of Beard and Pruppacher, and the droplet          *
                C *      acceleration components from the equations of motion.  *
                C *                                                             *
                C *  INPUT VARIABLES :                                          *
                C *                X,Y    - current droplet positions           *
                C *                VX,VY, - current droplet speeds              *
                C *                VEL    - free stream air speed               *
                C *                CYLRAD - cylinder radius                     *
                C *                DROPDI - droplet diameter                    *
                C *                STOKES - logical flag for Stokes flow        *
                C *                                                             *
                C *  OUTPUT VARIABLES :                                         *
                C *                DVX,DVY - current droplet accelerations      *
                C *                                                             *
                C *  OTHER VARIABLES :                                          *
                C *                UX,UY   - current air speed components       *
                C *                DROPRE  - droplet Reynolds number            *
                C *                DRAGCO  - drag coefficient                   *
                C *                COEFF   - combined cefficients of the        *
                C *                          equations of motion                *
                C *                DENOM   - denominator of air speed           *
                C *                          equations                          *
                C *                                                             *
                C ***************************************************************
                C
ISN       1     420.            SUBROUTINE ITERAT(X,Y,VX,VY,VEL,CYLRAD,DROPDI,STOKES,DVX,DVY)
                421.      C
ISN       2     422.            DOUBLE PRECISION X,Y,VX,VY,DVX,DVY,VEL,CYLRAD,DROPDI,
                423.           +                 DROPRE,DRAGCO,COEFF,UX,UY,DENOM
                424.      C
ISN       3     425.            LOGICAL           STOKES
                426.      C
                427.      C  Calculate air speeds
                428.      C
ISN       4     429.            DENOM = ((X * X) + (Y * Y)) ** 2
ISN       5     430.            UX    = VEL * (1 + ((CYLRAD**2) * (((Y * Y) - (X * X))/DENOM)))
ISN       6     431.            UY    = DABS(2 * VEL * (CYLRAD**2) * ((X * Y) / DENOM))
                432.      C
                433.      C Calculate Reynolds number and drag coefficient
                434.      C
ISN       7     435.            DROPRE    = DROPDI * DSQRT(((VX - UX) ** 2) + ((VY - UY)
                436.           +              ** 2)) / 1.262E-5
ISN       8     437.            IF (STOKES) THEN
ISN       9     438.               DRAGCO = 24.0 / DROPRE
ISN      10     439.            ELSE IF (DROPRE .LT. 2.0) THEN
ISN      11     440.               DRAGCO = (24.0 / DROPRE) + (2.45 / (DROPRE ** 0.045))
ISN      12     441.            ELSE IF (DROPRE .LT. 21.0) THEN
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN -          DATE: AUG 07, 1986    TIME: 10:44:54      NAME: ITERAT  PAGE:   11

                          *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*

ISN       13      442.              DRAGCO = (24.0 / DROPRE) + (2.76 / (DROPRE ** 0.198))
ISN       14      443.            ELSE IF (DROPRE .GT. 200.0) THEN
ISN       15      444.              DRAGCO = (24.0 / DROPRE) + (4.73 / (DROPRE ** 0.37))
                  445.         +                + (6.24E-3 * (DROPRE ** 0.38))
ISN       16      446.            ELSE
ISN       17      447.              DRAGCO = (24.0 / DROPRE) + (4.536 / (DROPRE ** 0.368))
ISN       18      448.            ENDIF
                  449.     C
                  450.     C  Calculate accelerations
                  453.     C
ISN       19      454.            COEFF = 9.92E-4 * (DRAGCO / DROPDI)
ISN       20      455.            DVX   = COEFF * DSQRT(((VX - UX) **2 ) + ((VY - UY) ** 2))
                  456.         +          * (VX - UX)
ISN       21      457.            DVY   = -COEFF * DSQRT(((VX - UX) ** 2) + ((VY - UY) ** 2))
                  458.         +          * (VY - UY)
                  459.     C
ISN       22      460.            RETURN
ISN       23      461.            END

*STATISTICS*   SOURCE STATEMENTS = 23, PROGRAM SIZE = 1380 BYTES, PROGRAM NAME = ITERAT    PAGE:   10.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 3 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                C ***************************************************************
                C *                                                             *
                C *  SUBROUTINE COLLID  -  Calculates the x,y position          *
                C *    of impact on the cylinder surface, position angle        *
                C *    of the impact point, and arc length measured from        *
                C *    the stagnation line.                                     *
                C *                                                             *
                C *  INPUT VARIABLES:                                           *
                C *               XOUT,  - last droplet position outside        *
                C *               YOUT     the cylinder                         *
                C *               XIN,   - first droplet position inside        *
                C *               YIN      the cylinder                         *
                C *               CYLRAD - cylinder radius                      *
                C *                                                             *
                C *  OUTPUT VARIABLES :                                         *
                C *               THETA  - position angle of impact point       *
                C *                        on the cylinder measured from        *
                C *                        the stagnation line                  *
                C *               ARC    - arc length from stagnation line      *
                C *                        to the impact point                  *
                C *                                                             *
                C *  OTHER VARIABLES :                                          *
                C *               SLOPE  - slope of the trajectory segment      *
                C *                        which crosses the cylinder surface   *
                C *                        i.e., between XOUT,YOUT and          *
                C *                        XIN,YIN                              *
                C *               A,B,C  - coefficients of the quadratic        *
                C *                        equation to solve for the intercept  *
                C *                        of the trajectory and cylinder       *
                C *                        surface                              *
                C *               IMPX,  - impact position on the surface       *
                C *               IMPY                                          *
                C *                                                             *
                C ***************************************************************
                C
ISN       1     497.            SUBROUTINE COLLID (XOUT,XIN,YOUT,YIN,CYLRAD,THETA,ARC)
                498.      C
ISN       2     499.            DOUBLE PRECISION XOUT,XIN,YOUT,YIN,CYLRAD,THETA,ARC,
                500.           +                 SLOPE,A,B,C,IMPX,IMPY
                501.      C
                502.      C  Calculate slope of trajectory at impact point
                503.      C
ISN       3     504.            SLOPE = (YIN - YOUT) / (XIN - XOUT)
ISN       4     505.            XOUT  = XOUT / CYLRAD
ISN       5     506.            XIN   = XIN / CYLRAD
ISN       6     507.            YOUT  = YOUT / CYLRAD
ISN       7     508.            YIN   = YIN / CYLRAD
                509.      C
                510.      C  Find impact point by solving quadratic equation with
                511.      C  coefficients as follows
                512.      C
ISN       8     513.            A     = (SLOPE * SLOPE) + 1.0
ISN       9     514.            B     = (2.0 * SLOPE * YOUT) - (2.0 * (SLOPE * SLOPE) * XOUT)
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       10    515.          C     = -(2.0 * SLOPE * XOUT * YOUT) + ((SLOPE * SLOPE) *
                516.         +          (XOUT * XOUT)) + (YOUT * YOUT) - 1.0
ISN       11    517.          IMPX = (-B - DSQRT((B * B) - (4 * A * C))) / (2 * A)
ISN       12    518.          IMPY = (SLOPE * (IMPX - XOUT)) + YOUT
                519.    C
                520.    C  Calculate impact point angle and arc length
                521.    C
ISN       13    522.          THETA = DATAN (IMPY / DABS(IMPX))
ISN       14    523.          ARC   = CYLRAD * THETA
                524.    C
ISN       15    525.          RETURN
ISN       16    526.          END

*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 840 BYTES, PROGRAM NAME = COLLID    PAGE:   12.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 4 ******
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:13:42      NAME: MAIN    PAGE:   14
SUMMARY OF MESSAGES AND STATISTICS FOR ALL COMPILATIONS
*STATISTICS*   SOURCE STATEMENTS = 69, PROGRAM SIZE = 2732 BYTES, PROGRAM NAME = MAIN    PAGE:     1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
*STATISTICS*   SOURCE STATEMENTS = 33, PROGRAM SIZE = 338354 BYTES, PROGRAM NAME = TRAJEC    PAGE:     7.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 2 ******
*STATISTICS*   SOURCE STATEMENTS = 23, PROGRAM SIZE = 1380 BYTES, PROGRAM NAME = ITERAT    PAGE:    10.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 3 ******
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 840 BYTES, PROGRAM NAME = COLLID    PAGE:    12.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 4 ******
******* SUMMARY STATISTICS ******* 0 DIAGNOSTICS GENERATED. HIGHEST SEVERITY CODE IS 0.
```

# Appendix II

## Program Listing: OMNICYL

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
1.    C        ********** Cylinder Rime Icing Model **********
2.    C
2.2   C        Karen J. Finstad  -  June, 1986
2.4   C
3.    C        This program will calculate and plot rime ice accretion
4.    C     layers on a cylinder, as well as calculate the mass and
5.    C     mean density of each layer. Local collision efficiencies
6.    C     for each point on the surface are calculated as a cosine
7.    C     function of the current local surface angle
8.    C     and scaled to the value of stagnation line collision
9.    C     efficiency (Beta zero), and maximum accretion angle
10.   C     (Alpha max).
11.   C
12.   C        Two forms of this function are used for droplet diameters
13.   C     above and below 20 microns. Beta zero and Alpha max are
14.   C     estimated from a non-linear regression fit to data computed
15.   C     by droplet trajectory program GRAZE.
16.   C
17.   C        Two optional procedures are available for treating
18.   C     the effects of droplet size spectra, when the assumption
19.   C     of a monodisperse population at the median volume
20.   C     diameter (mvd) becomes inaccurate (roughly, when the
21.   C     mvd is less than 50 micron, for ground level wind speeds):
22.   C           1. the user may enter weighting factors for each
23.   C     of 18 size bins of width 10 microns, or
24.   C           2. use a "simulated spectrum" curve for the local
25.   C     collision efficiency (l.c.e.), which differs principally
26.   C     from the cosine approximation by the addition of a "tail"
27.   C     onto the maximum accretion limit. The length of this
28.   C     tail is a function of the mvd. Again, there is a
29.   C     slightly different form of the curve for mvd above and
30.   C     below 20 microns.
31.   C        The first of these options is recommended only when
32.   C     three or less layers are to be accreted, since the weighted
33.   C     averaging procedure eventually produces first-order
34.   C     discontinuities in the l.c.e. curve, and thus on the
35.   C     modelled ice surface.
36.   C
37.   C        Local density is calculated from an empirically-
38.   C     derived function of local surface slope.
39.   C
40.   C        The cylinder may be rotated, clockwise, after each
41.   C     layer is accreted.The rotation increment is included
42.   C     in the input parameter list. When this option is
43.   C     chosen, (i.e., if the input rotation increment is greater
44.   C     than zero), the l.c.e. and local density are calculated
45.   C     as a function of position angle, instead of surface angle.
46.   C     This somewhat less accurate method is necessary to avoid
47.   C     problems when accreting over top of the previous layers'
48.   C     accretion limit.
49.   C        Also please note that rotation of more than 90 degrees
50.   C     in total is not possible in this version.
51.   C
```

```
      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
   52.  C        The output consists of a plot description file written into
   53.  C     -pdf attached to unit 8, and a mass and density table written
   54.  C     into -mass attached to unit 9. (These are temporary files.)
   55.  C
   56.  C        The plot description file is produced using the system
   57.  C     subroutine library *PLOTLIB. All of the plotting is done
   58.  C     in a single program subroutine, so that other available
   59.  C     plotting routines may be substituted if desired.
   60.  C        Please note: in order to run *PLOTLIB subroutines (or any
   61.  C     subroutines which have not been compiled in Fortran77 and
   62.  C     which contain character strings or variables in their
   63.  C     argument list), in conjunction with this program, (or any
   64.  C     Fortran77 program) the command @PROCESS(list of subroutines)
   65.  C     MUST appear in ROW 1, COLUMN 1 if the subroutines are
   66.  C     called directly from MAIN. If they are called from
   67.  C     another subroutine, the @PROCESS command must be placed
   68.  C     in COLUMN 1 OF THE LINE IMMEDIATELY FOLLOWING THE END
   69.  C     STATEMENT OF THE PREVIOUS ROUTINE.
   70.  C
   71.  C        This code will also run on the IBM Professional Fortran
   72.  C     compiler for PC's and PC compatibles. A math co-processor
   73.  C     and graphics capability is required. (Graphics routines
   74.  C     must be supplied). A 6-layer accretion with rotation
   75.  C     takes about 90 seconds on a standard PC. Storage needed
   76.  C     is about 370,000 bytes.
   77.  C
   78.  C        To compile (on MTS), first empty -mass and -pdf. Then
   79.  C     $RUN *FORTRANVS SCARDS=OMNIC SPUNCH=CYLRUN.
   80.  C
   81.  C        To run, enter $RUN CYLRUN+*PLOTLIB. You will be prompted
   82.  C     for the following input parameters:
   83.  C
   84.  C        median volume droplet DIAMETER in microns
   85.  C
   86.  C        y or n to enter full spectrum, followed by weights,
   87.  C          entered individually
   88.  C
   89.  C        y or n to simulated spectrum option
   90.  C
   91.  C        cylinder RADIUS in metres
   92.  C
   93.  C        free stream air speed in metres per second
   94.  C
   95.  C        liquid water content in kg per cubic metre
   96.  C
   97.  C        time step per layer in seconds
   98.  C
   99.  C        total number of layers
  100.  C
  101.  C        rotation increment in degrees
  102.  C
  103.  C
  104.  C     ********************** Subprograms **********************
  105.  C
  106.  C
  107.  C     INTERNAL:
```

```
*....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
C
C       SUBROUTINE DRAW - Plots cylinder profile and all
C     accreted ice layers with appropriate legend.
C
C       DOUBLE PRECISION FUNCTION MONBET - Estimates local
C     collision efficiency as a function of surface slope for
C     a monodisperse droplet population at the mvd. The
C     function has two forms for use with mvd smaller or
C     larger than than 20 microns.
C
C       SUBROUTINE PARAM - Estimates the stagnation line
C     collision efficiency and maximum accretion angle
C     for bare cylinders at atmospheric pressure of 100 kPa
C     from a non-linear regression fit to data derived from
C     droplet trajectory calculations.
C
C       SUBROUTINE ROTAT1 - When the rotation increment
C     is non-zero, this subroutine calculates the array
C     of position angles for the current surface, to replace
C     surface angles in the array ALPHA.
C
C       SUBROUTINE ROTAT2 - shifts the array indices of
C     the current LAYRX and LAYRY arrays to face the new
C     free stream direction, which has been rotated
C     according to the rotation increment.
C
C       DOUBLE PRECISION FUNCTION SPCBET - Estimates local
C     collision efficiency as a function of surface slope
C     which simulates the effect of a typical droplet size
C     spectrum centred on the mvd. The function has two forms
C     for use with mvd smaller or larger than 20 microns.
C
C       SUBROUTINE WTBET - Calculates weighted average of
C     local collision efficiency estimates from FUNCTION
C     MONBET, for droplet spectrum weights in 18 10-micron
C     size bins.
C
C
C     EXTERNAL:
C
C       *PLOTLIB  -  The subroutines PLOTS, METRIC, PLOT, AXIS2,
C     LINE, SYMBOL and NUMBER (called from SUBROUTINE DRAW) are
C     found in the public system subroutine library *PLOTLIB.
C     (See note above regarding external subroutines in Fortran77).
C
C     ****************** Variable Dictionary **********************
C
C       ALPHA       - array of angles between surface normal
C                     and the free stream direction
C
C       ALPHM       - Alpha max, the maximum impingement angle
C
C       AM          - array of maximum impingement angles for
C                     each size category of the droplet size
C                     spectrum
C
```

```
*....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      C
      C     AREA          - cross sectional area between
      C                     adjacent points in the current layer
      C                     (dimensioned in metres)
      C
      C     BETA          - local collision efficiency
      C
      C     BZERO         - stagnation line value of the local collision
      C                     efficiency
      C
      C     BZ            - array of Bzero values for each size
      C                     category of the droplet size spectrum
      C
      C     CYLRAD        - radius of cylinder in metres (INPUT)
      C
      C     CYLX,CYLY     - arrays containing non - dimensional x and
      C                     y coordinates of the cylinder surface
      C
      C     DI            - array of 18 10-micron wide droplet
      C                     diameter size bins
      C
      C     DIAG          - cross diagonal of the quadrilateral
      C                     area used in mass calculations
      C
      C     DROPDI        - droplet diameter in microns (INPUT)
      C
      C     HALFPI        - pi/2 (Fortran77 constant PARAMETER)
      C
      C     LAREA         - cross sectional area of the current layer
      C                     in sq. metres
      C
      C     LAYERS        - integer number of layers
      C
      C     LAYRX,LAYRY   - array containing non - dimensional x and y
      C                     coordinates of the accretion surfaces for
      C                     up to 30 accretion layers
      C
      C     LDENS         - mean density of the current layer in kg/cubic m
      C
      C     LMASS         - mass of the current layer in kg per unit length
      C
      C     LWC           - liquid water content in kg per cubic metre
      C                     (INPUT)
      C
      C     L1,L2,L3      - dimensioned sides of the quadrilateral area
      C     L4              defined by two adjacent points on the
      C                     current surface and two on the surface below
      C
      C     MASS          - array of masses per unit length for areas
      C                     defined by adjacent points, in kg
      C
      C     PI            - pi (Fortran77 constant PARAMETER)
      C
      C     R             - temporary variable for storing real result
      C                     during conversion of the rotation increment
      C                     an integer number of surface points
      C
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                220.    C       RHO         - array of local densities in kg/cubic metre
                221.    C
                222.    C       ROTN        - no. of surface points in a rotation increment
                223.    C
                224.    C       RSPEED      - rotation increment per layer in radians (INPUT)
                225.    C
                226.    C       SIMSPC      - logical variable, true if simulated spectrum
                227.    C                     option is chosen
                228.    C
                229.    C       SIMOPT      - character variable for reading in response to
                230.    C                     simulated spectrum option query (INPUT)
                231.    C
                232.    C       SLOPE       - array of local surface slopes calculated
                233.    C                     at each surface point
                234.    C
                235.    C       SMDROP      - logical flag for droplet size, which determines
                236.    C                     function used to calculate local c.e.
                237.    C
                238.    C       SPCOPT      - character variable for reading in response
                239.    C                     to spectrum weighting option query (INPUT)
                240.    C
                241.    C       SPEC        - logical variable, true if spectrum weighting
                242.    C                     option is chosen
                243.    C
                244.    C       S1,S2       - quantities used to determine the quadrilaterals
                245.    C                     for the layer area calculation
                246.    C
                247.    C       TAILEN      - filter length for optional smoothing filter
                248.    C                     applied to c. e. curve
                249.    C
                250.    C       TAU         - accretion time step in seconds (INPUT)
                251.    C
                252.    C       THICK       - array of ice thicknesses normal to surface
                253.    C                     (non - dimensional)
                254.    C
                255.    C       TMASS       - total mass of the accretion
                256.    C
                257.    C       VEL         - free stream air speed in m per sec (INPUT)
                258.    C
                259.    C       WGHT        - array of droplet spectrum volume percentages
                260.    C                     for each size bin (INPUT)
                261.    C
                262.    C
                263.    C ***********************************************************
                264.    C
ISN         1   265.            DOUBLE PRECISION ALPHA(600),CYLX(1202),CYLY(1202),
                266.         +                   LAYRX(30,600),LAYRY(30,600),
                267.         +                   RHO(600),SLOPE(600),THICK(600),
                268.         +                   DI(18),WGHT(18),BZ(18),AM(18)
                269.    C
ISN         2   270.            DOUBLE PRECISION ALPHM,AREA,BETA,BZERO,CYLRAD,DIAG,
                271.         +                   DROPDI,LAREA,LDENS,LMASS,LWC,L1,L2,
                272.         +                   L3,L4,MASS,RSPEED,S1,S2,TAILEN,TAU,
                273.         +                   TMASS,VEL
                274.    C
ISN         3   275.            REAL             R
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                276.    C
ISN       4     277.          INTEGER          LAYERS,ROTN
                278.    C
ISN       5     279.          CHARACTER*1      SIMOPT,SPCOPT
                280.    C
ISN       6     281.          LOGICAL          SIMSPC,SMDROP,SPEC
                282.    C
ISN       7     283.          PARAMETER (PI = 3.1415926535897932,HALFPI = PI/2)
                284.    C
ISN       8     285.          COMMON CYLX,CYLY,LAYRX,LAYRY
                286.    C
                287.    C  Fill array of droplet diameter bins
                288.    C
ISN       9     289.               DATA (DI(I),I=1,18)/5.E-6,15.E-6,25.E-6,35.E-6,45.E-6,
                290.           +           55.E-6,65.E-6,75.E-6,85.E-6,95.E-6,105.E-6,
                291.           +           115.E-6,125.E-6,135.E-6,145.E-6,155.E-6,
                292.           +           165.E-6,175.E-6/
                293.    C
                294.    C  Prepare mass table and plot description files.
                295.    C
ISN      10     296.          OPEN (UNIT = 8, FILE = '-MASS', STATUS = 'OLD')
ISN      11     297.          OPEN (UNIT = 9, FILE = '-PDF', STATUS = 'OLD')
                298.    C
                299.    C  Set up headings for mass and density table in -MASS.
                300.    C
ISN      12     301.          WRITE (8,110)
ISN      13     302.        110  FORMAT (///,15X,' Mass of Ice Accreted',//,5X,' Layer no.',
                303.           +         7X,' Mass',7X,' Mean Density',//,
                304.           +         20X,' (kg/m)',9X,' (kg/cu.m)',//,
                305.           +         ' ************************************************')
                306.    C
                307.    C   Write program title and intro to screen.
                308.    C
ISN      14     309.          WRITE (6,120)
ISN      15     310.        120  FORMAT (/////,' *************  CYLINDER RIME ICING MODEL  *****',
                311.           +         ' ************',//,20X,' Version Feb. 17, 1986',//,
                312.           +         22X,' K. J. Finstad',/,19X,
                313.           +         ' Meteorology Division',/,15X,' The University of Alberta'
                314.           +         ,//,' Please enter parameters to begin accretion: ',//)
                315.    C
                316.    C    Prompt user for input parameters.
                317.    C
ISN      16     318.          WRITE (6,*) 'Median Volume Droplet DIAMETER in microns? '
ISN      17     319.          READ  (5,*) DROPDI
                320.    C
                321.    C  Set droplet size flag
                322.    C
ISN      18     323.          SMDROP = .FALSE.
ISN      19     324.          IF (DROPDI .LT. 20) SMDROP = .TRUE.
ISN      21     325.          DROPDI = DROPDI*1.0E-6
                326.    C
                327.    C   Set flags for requested treatment of droplet spectrum.
                328.    C   If full spectrum weighting is requested, input volume
                329.    C   weights for each size bin.
                330.    C
ISN      22     331.          WRITE (6,*)
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN      23    332.          WRITE (6,*) 'Do you wish to enter full droplet spectrum? '
ISN      24    333.          READ  (5,9) SPCOPT
ISN      25    334.        9 FORMAT (1A1)
ISN      26    335.          SPEC = .FALSE.
ISN      27    336.          IF ((SPCOPT .EQ. 'y') .OR. (SPCOPT .EQ. 'Y')) THEN
ISN      28    337.             SPEC = .TRUE.
ISN      29    338.             WRITE (6,*)
ISN      30    339.             WRITE (6,*) 'Enter weight for the following size ranges:'
ISN      31    340.             DO 13 I=1,18
ISN      32    341.                WRITE (6,*)
ISN      33    342.                WRITE (6,*) DI(I)/1.0E-6, 'microns'
ISN      34    343.                READ (5,*) WGHT(I)
ISN      35    344.       13    CONTINUE
ISN      36    345.          ELSE
ISN      37    346.             WRITE (6,*)
ISN      38    347.             WRITE (6,*) 'Do you wish simulated spectrum weighting? '
ISN      39    348.             READ  (5,8) SIMOPT
ISN      40    349.          8  FORMAT (1A1)
ISN      41    350.          ENDIF
               351.    C
               352.    C  For simulated spectrum, calculate 'tail' length from
               353.    C  median volume diameter.
               354.    C
ISN      42    355.          IF ((SIMOPT .EQ. 'Y') .OR. (SIMOPT .EQ. 'y')) THEN
ISN      43    356.            SIMSPC = .TRUE.
ISN      44    357.            IF (SMDROP) THEN
ISN      45    358.               TAILEN = 0.52 * DDEXP(-DROPDI / 7.0E-5)
ISN      46    359.            ELSE
ISN      47    360.               TAILEN = 0.37 * DDEXP(-DROPDI / 7.0E-5)
ISN      48    361.            ENDIF
ISN      49    362.          ELSE
ISN      50    363.            SIMSPC = .FALSE.
ISN      51    364.            TAILEN = 0.0
ISN      52    365.          ENDIF
               366.    C
ISN      53    367.          WRITE (6,*)
ISN      54    368.          WRITE (6,*) 'Cylinder RADIUS, in metres? '
ISN      55    369.          READ  (5,*) CYLRAD
               370.    C
ISN      56    371.          WRITE (6,*)
ISN      57    372.          WRITE (6,*) 'Free stream speed in m / sec? '
ISN      58    373.          READ  (5,*) VEL
               374.    C
ISN      59    375.          WRITE (6,*)
ISN      60    376.          WRITE (6,*) 'Liquid water content in kg / cubic metre? '
ISN      61    377.          READ  (5,*) LWC
               378.    C
ISN      62    379.          WRITE (6,*)
ISN      63    380.          WRITE (6,*) 'Time step per layer in seconds? '
ISN      64    381.          READ  (5,*)  TAU
               382.    C
ISN      65    383.          WRITE (6,*)
ISN      66    384.          WRITE (6,*) 'Total number of layers? '
ISN      67    385.          READ  (5,*) LAYERS
               386.    C
               387.    C   Bare cylinder will be layer 1, so add 1 to LAYERS
```

```
                                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                    388.    C
ISN         68      389.            LAYERS = LAYERS + 1
                    390.    C
ISN         69      391.            WRITE (6,*)
ISN         70      392.            WRITE (6,*) 'Rotation speed in degrees per layer? '
ISN         71      393.            READ  (5,*) RSPEED
                    394.    C
                    395.    C  Convert rotation increment to radians, and calculate
                    396.    C  approx. number of surface points in rotation increment
                    397.    C
ISN         72      398.            RSPEED = (RSPEED * 2.0 * PI) / 360.0
ISN         73      399.            R      = (RSPEED / 5.235987E-3)
ISN         74      400.            ROTN   = NINT(R)
                    401.    C
                    402.    C
                    403.    C  Call Subroutine Param to calculate Beta zero and alpha max
                    404.    C  for the mvd. If spectrum weighting option has been chosen,
                    405.    C  call Param for each size bin.
                    406.    C
ISN         75      407.            CALL PARAM (DROPDI,VEL,CYLRAD,BZERO,ALPHM)
ISN         76      408.            IF (SPEC) THEN
ISN         77      409.               DO 12 I=1,18
ISN         78      410.                  IF (WGHT(I) .NE. 0.0) THEN
ISN         79      411.                     CALL PARAM(DI(I),VEL,CYLRAD,BZ(I),AM(I))
ISN         80      412.                  ELSE
ISN         81      413.                     BZ(I) = 0.0
ISN         82      414.                     AM(I) = 0.0
ISN         83      415.                  ENDIF
                    416.    C
                    417.    C  Make sure accretion limit is revised to reflect the
                    418.    C  largest droplets present.
                    419.    C
ISN         84      420.                  IF (AM(I) .GT. AM(I-1)) ALPHM = AM(I)
ISN         86      421.           12     CONTINUE
ISN         87      422.            ENDIF
                    423.    C
                    424.    C  Fill array of points defining the initial  surface of the
                    425.    C  cylinder in non-dimensional coordinates. The complete
                    426.    C  surface contains 1200 points spaced equally in x.
                    427.    C
                    428.    C
ISN         88      429.            DO 100 I=601,1200
ISN         89      430.               CYLX(I) = (DBLE(I-601) * 3.33333E-3)
ISN         90      431.               CYLY(I) = DSQRT(1.0 - ((1.0 - CYLX(I))*(1.0 - CYLX(I))))
ISN         91      432.           100 CONTINUE
ISN         92      433.            DO 150 I = 1,600
ISN         93      434.               CYLX(I) = 2.0 - (DBLE(I) * 3.33333E-3)
ISN         94      435.               CYLY(I) = -DSQRT(1. - ((1. - CYLX(I)) * (1. - CYLX(I))))
ISN         95      436.           150 CONTINUE
                    437.    C
                    438.    C  Copy front half of  surface to array which will contain
                    439.    C  all of the layers.
                    440.    C
ISN         96      441.            DO 200 I = 1,600
ISN         97      442.              LAYRX(1,I) = CYLX(299 + I)
ISN         98      443.              LAYRY(1,I) = CYLY(299 + I)
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN      99     444.     200   CONTINUE
               445.    C
               446.    C  Initialize total mass and begin loop to accrete ice layers.
               447.    C
ISN     100     448.          TMASS = 0.0
ISN     101     449.          DO 300 J = 2,LAYERS
               450.    C
               451.    C  Calculate slope and surface angle from finite differences for
               452.    C  each point on current surface. If rotation increment is
               453.    C  non-zero, call subroutine to calculate slopes and
               454.    C  position angles instead.
               455.    C
ISN     102     456.          IF (ROTN .GT. 0) THEN
ISN     103     457.             CALL ROTAT1(J,RSPEED,SLOPE,ALPHA)
ISN     104     458.          ELSE
ISN     105     459.            DO 400 I = 2,599
ISN     106     460.               SLOPE(I) = (LAYRX(J-1,I+1) - LAYRX(J-1,I-1)) /
               461.      +                   DABS(LAYRY(J-1,I+1) - LAYRY(J-1,I-1))
ISN     107     462.               IF (LAYRY(J-1,I+1) .LT. LAYRY(J-1,I-1)) THEN
ISN     108     463.                  ALPHA(I) = DABS(PI - DATAN(SLOPE(I)))
ISN     109     464.               ELSE
ISN     110     465.                  ALPHA(I) = DABS(DATAN(SLOPE(I)))
ISN     111     466.               ENDIF
ISN     112     467.     400     CONTINUE
ISN     113     468.          ENDIF
               469.    C
               470.    C  Calculate for each point local density, collision efficiency and
               471.    C  ice thickness. The method of calculation for Beta depends on
               472.    C  the values of SMDROP, SPEC and SIMSPC.
               473.    C
ISN     114     474.            DO 500 I = 2,599
ISN     115     475.               RHO(I) = 890. *
               476.      +              ( (1.0 - .143 * (ALPHA(I)/(ALPHM+TAILEN)) -
               477.      +                .246 * ((ALPHA(I)/(ALPHM+TAILEN)) ** 2) -
               478.      +                .309 * ((ALPHA(I)/(ALPHM+TAILEN)) ** 3))
               479.    C
               480.    C  Make sure density does not become too small
               481.    C
ISN     116     482.               IF (RHO(I) .LT. 175.) RHO(I) = 175.
               483.    C
               484.    C  Calculate local collision efficiency
               485.    C
ISN     118     486.               IF (SPEC) THEN
ISN     119     487.                  CALL WTBET(ALPHA(I),SMDROP,BZ,AM,WGHT,BETA)
ISN     120     488.               ELSE IF (SIMSPC) THEN
ISN     121     489.                  BETA = SPCBET(ALPHA(I),SMDROP,BZERO,ALPHM,TAILEN)
ISN     122     490.               ELSE
ISN     123     491.                  BETA = MONBET(ALPHA(I),SMDROP,BZERO,ALPHM)
ISN     124     492.               ENDIF
               493.    C
               494.    C  Calculate local ice thickness perpendicular to surface
               495.    C  (in non-dimensional units)
               496.    C
ISN     125     497.               THICK(I)   = BETA * VEL * TAU * LWC / (RHO(I) * CYLRAD)
               498.    C
               499.    C  Define new layer surface coordinates.
```

```
                                *....*....1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                    500.    C
ISN       126       501.                IF (LAYRY(J-1,I) .LT. LAYRY(J-1,I-1)) THEN
ISN       127       502.                   LAYRX(J,I) = LAYRX(J-1,I) + (THICK(I) *
                    503.         +                      DCOS(DATAN(SLOPE(I))))
ISN       128       504.                ELSE
ISN       129       505.                   LAYRX(J,I) = LAYRX(J-1,I) - (THICK(I) *
                    506.         +                      DCOS(DATAN(SLOPE(I))))
ISN       130       507.                ENDIF
ISN       131       508.                LAYRY(J,I) = LAYRY(J-1,I) + (THICK(I) *
                    509.         +                    DSIN(DATAN(SLOPE(I))))
ISN       132       510.              500    CONTINUE
ISN       133       511.             LAYRX(J,1)   = LAYRX(J-1,1)
ISN       134       512.             LAYRX(J,600) = LAYRX(J-1,600)
ISN       135       513.             LAYRY(J,1)   = LAYRY(J-1,1)
ISN       136       514.             LAYRY(J,600) = LAYRY(J-1,600)
                    515.    C
                    516.    C  Calculate mass and mean density of the layer from
                    517.    C  the sum of areas between surface points on successive
                    518.    C  layers. First, initialize layer mass and area.
                    519.    C
ISN       137       520.             LMASS = 0.0
ISN       138       521.             LAREA = 0.0
ISN       139       522.             DO 560 I = 2,598
ISN       140       523.                IF ((THICK(I) .GT. 0.0).AND.(THICK(I+1) .GT. 0.0)) THEN
ISN       141       524.                   L4 = THICK(I) * CYLRAD
ISN       142       525.                   L2 = THICK(I+1) * CYLRAD
ISN       143       526.                   L1 = DSQRT((LAYRX(J-1,I+1) - LAYRX(J-1,I))**2 + (LAYRY
                    527.         +                (J-1,I+1) - LAYRY(J-1,I))**2) * CYLRAD
ISN       144       528.                   L3 = DSQRT((LAYRX(J,I+1) - LAYRX(J,I))**2 + (LAYRY
                    529.         +                (J,I+1) - LAYRY(J,I))**2) * CYLRAD
ISN       145       530.                   DIAG = DSQRT((LAYRX(J,I+1)-LAYRX(J-1,I))**2 + (LAYRY
                    531.         +                  (J,I+1) - LAYRY(J-1,I))**2) * CYLRAD
ISN       146       532.                   S1   = 0.5 * (DIAG+L1+L2)
ISN       147       533.                   S2   = 0.5 * (DIAG+L3+L4)
ISN       148       534.                   AREA = DSQRT(S1 * (S1-DIAG) * (S1-L1) * (S1-L2)) +
                    535.         +                     DSQRT(S2 * (S2-DIAG) * (S2-L3) * (S2-L4))
ISN       149       536.                   MASS = ((RHO(I) + RHO(I+1)) / 2.0) * AREA
ISN       150       537.                   LMASS  = LMASS + MASS
ISN       151       538.                   LAREA  = LAREA + AREA
ISN       152       539.                ENDIF
ISN       153       540.              560    CONTINUE
                    541.    C
                    542.    C  Calculate mean density.
                    543.    C
ISN       154       544.             LDENS = LMASS/LAREA
                    545.    C
                    546.    C  Write results to mass table.
                    547.    C
ISN       155       548.             WRITE (8,570) J-1,LMASS,LDENS
ISN       156       549.           570    FORMAT (//,8X,I2,7X,F9.5,7X,F9.5)
                    550.    C
                    551.    C  Add mass of current layer to total mass.
                    552.    C
ISN       157       553.             TMASS = TMASS + LMASS
                    554.    C
                    555.    C  Call subroutine to rotate array indices if required.
```

```
                      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                556.    C
ISN      158    557.              IF (ROTN .GT. 0) CALL ROTAT2(J,ROTN)
                558.    C
ISN      160    559.      300   CONTINUE
                560.    C
                561.    C  Write total mass accreted to table
                562.    C
ISN      161    563.           WRITE (8,590) TMASS
ISN      162    564.      590   FORMAT (8X,//,' Total mass accreted per metre:',F9.5)
                565.    C
                566.    C  Call subroutine to plot cylinder and accreted layers
                567.    C
ISN      163    568.           CALL DRAW(ALPHM,BZERO,CYLRAD,DROPDI,LAYERS,LWC,RSPEED,
                569.          +          SIMSPC,SPEC,TAILEN,TAU,VEL)
ISN      164    570.      999   STOP
ISN      165    571.           END

*STATISTICS*   SOURCE STATEMENTS = 161, PROGRAM SIZE = 27174 BYTES, PROGRAM NAME = MAIN    PAGE:    1.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 1 ******
```

REQUESTED OPTIONS (PROCESS): SC(AXIS2,SYMBOL)

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                573.    C
                574.    ****************************************************************
                575.    *  SUBROUTINE DRAW - to plot cylinder and all accretion       *
                576.    *                    layers, with approriate information     *
                577.    *                    in the legend. This subroutine uses     *
                578.    *                    the plotting library *PLOTLIB           *
                579.    *                                                            *
                580.    *  Input Variables : as defined for Main program ;           *
                581.    *                    ALPHM,BZERO,CYLRAD,DROPDI,LAYERS,LWC,   *
                582.    *                    RSPEED,SIMSPC,SPEC,TAILEN,TAU,VEL       *
                583.    *                                                            *
                584.    *  Common Variables : CYLX,CYLY,LAYRX,LAYRY                  *
                585.    *                                                            *
                586.    *  Other Variables : PLOTX,PLOTY - arrays containing x,y     *
                587.    *                    plotting data and scaling information   *
                588.    *                    for accretion layer currently being     *
                589.    *                    plotted                                 *
                590.    *                                                            *
                591.    ****************************************************************
                592.    C
ISN        1    593.          SUBROUTINE DRAW(ALPHM,BZERO,CYLRAD,DROPDI,LAYERS,
                594.         +                LWC,RSPEED,SIMSPC,SPEC,TAILEN,TAU,VEL)
                595.    C
ISN        2    596.          DOUBLE PRECISION CYLX(1202),CYLY(1202),LAYRX(30,600),
                597.         +                 LAYRY(30,600),PLOTX(602),PLOTY(602)
                598.    C
ISN        3    599.          DOUBLE PRECISION ALPHM,BZERO,CYLRAD,DROPDI,LWC,RSPEED,
                600.         +                 TAILEN,TAU,VEL
                601.    C
ISN        4    602.          INTEGER          LAYERS
                603.    C
ISN        5    604.          LOGICAL          SIMSPC,SPEC
                605.    C
ISN        6    606.          COMMON CYLX,CYLY,LAYRX,LAYRY
                607.    C
                608.    C  To plot cylinder surface, form lower surface array,
                609.    C  and place scaling data at the end of coordinate arrays.
                610.    C
ISN        7    611.          CYLX(1201) = -6.0
ISN        8    612.          CYLX(1202) =  1.0
ISN        9    613.          CYLY(1201) = -3.0
ISN       10    614.          CYLY(1202) =  1.0
ISN       11    615.          PLOTX(601) = -6.0
ISN       12    616.          PLOTX(602) =  1.0
ISN       13    617.          PLOTY(601) = -3.0
ISN       14    618.          PLOTY(602) =  1.0
                619.    C
                620.    C  Open plotting library
                621.    C
ISN       15    622.          CALL PLOTS
ISN       16    623.          CALL METRIC(0)
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                624.    C
                625.    C  Draw 6 by 9 inch box
                626.    C
ISN      17     627.          CALL PLOT(2.0,2.0,-3)
ISN      18     628.          CALL PLOT(0.0,6.0,3)
ISN      19     629.          CALL PLOT(9.0,6.0,2)
ISN      20     630.          CALL PLOT(9.0,0.0,2)
                631.    C
                632.    C  Label axes and draw tics
                633.    C
ISN      21     634.          CALL AXIS2(0.0,0.0,'X/CYLRAD',-8,9.0,0.0,-6.0,1.0,1.0)
ISN      22     635.          CALL AXIS2(0.0,0.0,'Y/CYLRAD',8,6.0,90.0,-3.0,1.0,0.5)
                636.    C
                637.    C  Write legends to top left corner
                638.    C
ISN      23     639.          CALL SYMBOL(1.1,5.4,0.10,'DROPLET MVD  = ',0.0,15)
ISN      24     640.          CALL NUMBER(2.6,5.4,0.10,DROPDI,0.0,07)
ISN      25     641.          CALL SYMBOL(3.5,5.4,0.10,'M',0.0,01)
ISN      26     642.          CALL SYMBOL(1.1,5.2,0.10,'CYL RADIUS   = ',0.0,15)
ISN      27     643.          CALL NUMBER(2.6,5.2,0.10,CYLRAD,0.0,04)
ISN      28     644.          CALL SYMBOL(3.5,5.2,0.10,'M',0.0,01)
ISN      29     645.          CALL SYMBOL(1.1,5.0,0.10,'AIR SPEED    = ',0.0,15)
ISN      30     646.          CALL NUMBER(2.6,5.0,0.10,VEL,0.0,02)
ISN      31     647.          CALL SYMBOL(3.5,5.0,0.10,'M / SEC',0.0,07)
ISN      32     648.          CALL SYMBOL(1.1,4.8,0.10,'LWC          = ',0.0,15)
ISN      33     649.          CALL NUMBER(2.6,4.8,0.10,LWC,0.0,05)
ISN      34     650.          CALL SYMBOL(3.5,4.8,0.10,'KG / M**3',0.0,09)
ISN      35     651.          CALL SYMBOL(1.1,4.6,0.10,'TIME STEP    = ',0.0,15)
ISN      36     652.          CALL NUMBER(2.6,4.6,0.10,TAU,0.0,02)
ISN      37     653.          CALL SYMBOL(3.5,4.6,0.10,'SEC',0.0,03)
ISN      38     654.          CALL SYMBOL(1.1,4.4,0.10,'BETA ZERO    = ',0.0,15)
ISN      39     655.          CALL NUMBER(2.6,4.4,0.10,BZERO,0.0,03)
ISN      40     656.          CALL SYMBOL(1.1,4.2,0.10,'ALPHA MAX    = ',0.0,15)
ISN      41     657.          CALL NUMBER(2.6,4.2,0.10,(ALPHM+TAILEN)*360./6.283,0.0,01)
ISN      42     658.          CALL SYMBOL(3.5,4.2,0.10,'DEGREES',0.0,07)
ISN      43     659.          CALL SYMBOL(1.1,4.0,0.10,'ROTN INCR    = ',0.0,15)
ISN      44     660.          CALL NUMBER(2.6,4.0,0.10,RSPEED*360./6.283,0.0,01)
ISN      45     661.          CALL SYMBOL(3.5,4.0,0.10,'DEGREES',0.0,07)
ISN      46     662.          CALL SYMBOL(1.1,3.8,0.10,'SPECTRUM:',0.0,09)
ISN      47     663.          IF (SPEC) THEN
ISN      48     664.             CALL SYMBOL(2.6,3.8,0.10,'YES',0.0,03)
ISN      49     665.          ELSE IF (SIMSPC) THEN
ISN      50     666.             CALL SYMBOL(2.6,3.8,0.10,'SIMULATED',0.0,09)
ISN      51     667.          ELSE
ISN      52     668.             CALL SYMBOL(2.6,3.8,0.10,'NO',0.0,02)
ISN      53     669.          ENDIF
                670.    C
                671.    C  Draw initial cylinder surface
                672.    C
ISN      54     673.          CALL LINE(CYLX,CYLY,1200,2,0)
                674.    C
                675.    C  Transfer each layer in turn to the array holding
                676.    C  the scaling parameters
                677.    C
ISN      55     678.          DO 610 J = 2,LAYERS
ISN      56     679.             DO 700 L = 1,600
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:29      NAME: DRAW    PAGE:   14

                         *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*

ISN       57      680.                  PLOTX(L) =  LAYRX(J,L)
ISN       58      681.                  PLOTY(L) =  LAYRY(J,L)
ISN       59      682.            700     CONTINUE
                  683.      C
                  684.      C  Draw accretion layer
                  685.      C
ISN       60      686.                  CALL LINE(PLOTX,PLOTY,600,2,0)
ISN       61      687.            610   CONTINUE
                  688.      C
                  689.      C  Close plotting library
                  690.      C
ISN       62      691.                  CALL PLOT(1.0,1.0,999)
ISN       63      692.                  RETURN
ISN       64      693.                  END

*STATISTICS*   SOURCE STATEMENTS = 64, PROGRAM SIZE = 12132 BYTES, PROGRAM NAME = DRAW    PAGE:   12.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 2 ******
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:30      NAME: MAIN    PAGE:    15

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED          NOTEST   SEQ
                      OPTIMIZE(0)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                          *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*

                    694.      C ************************************************************
                    695.      C *  FUNCTION MONBET - Calculates local c. e.                 *
                    696.      C *   estimate for monodisperse droplet populations           *
                    697.      C *   at mvd either above or below 20 microns                 *
                    698.      C *   using a cosine curve scaled to the stagnation           *
                    699.      C *   line c. e. and the maximum accretion angle.             *
                    700.      C *                                                           *
                    701.      C *  Input Variables : as defined for main program;          *
                    702.      C *                    ALPHM,BZERO,ALPHA,SMDROP               *
                    703.      C *                                                           *
                    704.      C ************************************************************
                    705.      C
ISN          1      706.            DOUBLE PRECISION FUNCTION MONBET(ALPHA,SMDROP,BZERO,ALPHM)
                    707.      C
ISN          2      708.            DOUBLE PRECISION BZERO,ALPHM,ALPHA
                    709.      C
ISN          3      710.            LOGICAL          SMDROP
                    711.      C
ISN          4      712.            PARAMETER        (HALFPI = 3.1415926535897932/2.)
                    713.      C
ISN          5      714.            IF (ALPHA .GE. ALPHM) THEN
ISN          6      715.               MONBET = 0.0
ISN          7      716.            ELSE IF (SMDROP) THEN
ISN          8      717.               MONBET = BZERO + (BZERO * (DSQRT(DCOS(ALPHA)) - 1.0)) /
                    718.           +                   (1.0 - DSQRT(DCOS(ALPHM)))
ISN          9      719.            ELSE
ISN         10      720.               MONBET = BZERO * (DCOS((ALPHA/ALPHM) * HALFPI))
ISN         11      721.            ENDIF
ISN         12      722.            RETURN
ISN         13      723.            END

*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 586 BYTES, PROGRAM NAME = MONBET    PAGE:    15.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 3 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                724.    C
                725.    C ***********************************************************
                726.    C *  Subroutine PARAM - Estimates Beta zero and alpha max  *
                727.    C *              from the input parameters according to     *
                728.    C *              an empirical fit to droplet trajectories   *
                729.    C *              calculated with program COLLISION.         *
                730.    C *                                                         *
                731.    C *  Input variables : as defined for Main program;        *
                732.    C *                    CYLRAD,DROPDI,VEL                    *
                733.    C *                                                         *
                734.    C *  Output variables : as defined for Main program;       *
                735.    C *                     ALPHM,BZERO                         *
                736.    C *                                                         *
                737.    C *  Other variables :                                     *
                738.    C *           RE       - free stream Reynolds number       *
                739.    C *           K        - Langmuir inertia parameter        *
                740.    C *           PHI      - Langmuir parameter                *
                741.    C *           TERM1,   - terms used in the fitting function *
                742.    C *           TERM2,                                       *
                743.    C *           TERM3                                        *
                744.    C *           A(2,12) - array containing constants for     *
                745.    C *                      the fitting functions            *
                746.    C *                                                         *
                747.    C ***********************************************************
                748.    C
ISN       1     749.          SUBROUTINE PARAM (DROPDI,VEL,CYLRAD,BZERO,ALPHM)
                750.    C
ISN       2     751.          DOUBLE PRECISION ALPHM,BZERO,CYLRAD,DROPDI,K,
                752.         +                 PHI,RE,TERM1,TERM2,TERM3,VEL
                753.    C
ISN       3     754.          DOUBLE PRECISION A(2,12)
                755.    C
ISN       4     756.          DATA (A(1,I), I=1,12)/1.215,-7.42D-3,-0.561,-0.644,-0.164,
                757.         +              3.05D-3,0.430,2.22,-0.45,-0.767,-0.806,-0.068/
ISN       5     758.          DATA (A(2,I), I=1,12)/2.327,-4.09D-3,-0.388,-0.593,-0.689,
                759.         +              8.5D-3,0.383,1.757,-0.298,-0.42,-0.96,-0.179/
                760.    C
                761.    C  Calculate free stream Reynolds number, phi and K assuming
                762.    C  water density is 1000, air density 1.323 and air
                763.    C  viscosity 1.669 in MKS units. These correspond to
                764.    C  an air temperature of -10 C and pressure 100 kPa.
                765.    C
                766.    C
ISN       6     767.          RE = DROPDI * VEL / 1.262E-5
ISN       7     768.          K  = (VEL * DROPDI * DROPDI * 3.329E6) / CYLRAD
ISN       8     769.          PHI = (RE * RE) / K
                770.    C
                771.    C  Calculate Beta zero and alpha max
                772.    C
ISN       9     773.          DO 10 I = 1,2
ISN      10     774.             TERM1 = (A(I,1) * (K ** A(I,2))*DEXP(A(I,3)*K**A(I,4)))
                775.         +           + A(I,5)
ISN      11     776.             TERM2 = A(I,6) *(PHI ** A(I,7))
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       12    777.            TERM3 = (A(I,8) * (K ** A(I,9))*DEXP(A(I,10)*K**A(I,11)))
                778.       +              + A(I,12)
ISN       13    779.            IF (I .EQ. 1) THEN
ISN       14    780.               BZERO = TERM1 - (TERM2 * TERM3)
ISN       15    781.            ELSE
ISN       16    782.               ALPHM = TERM1 - (TERM2 * TERM3)
ISN       17    783.            ENDIF
ISN       18    784.         10 CONTINUE
                785.    C
                786.    C  Check for extreme values of K
                787.    C
ISN       19    788.            IF (K .LT. 0.17) THEN
ISN       20    789.               ALPHM = 0.0
ISN       21    790.               BZERO = 0.0
ISN       22    791.            ELSE IF (K .GT. 1000.) THEN
ISN       23    792.               ALPHM = 1.56
ISN       24    793.               BZERO = 0.99
ISN       25    794.            ENDIF
ISN       26    795.            RETURN
ISN       27    796.            END

*STATISTICS*   SOURCE STATEMENTS = 27, PROGRAM SIZE = 1362 BYTES, PROGRAM NAME = PARAM    PAGE:   16.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 4 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                797.    C
                798.    C ************************************************************
                799.    C *  SUBROUTINE ROTAT1 - calculates position angles and     *
                800.    C *                      slopes for each point on the       *
                801.    C *                      current surface. Position angles   *
                802.    C *                      are with respect to the current    *
                803.    C *                      free stream direction.             *
                804.    C *                                                         *
                805.    C *  Input Variables : as defined for Main program;        *
                806.    C *                     J,RSPEED                            *
                807.    C *                                                         *
                808.    C *  Output Variables :                                    *
                809.    C *                     ALPHA - array of position angles    *
                810.    C *                             measures w.r.t. center of   *
                811.    C *                             cylinder and free stream    *
                812.    C *                     as defined for Main program;        *
                813.    C *                     SLOPE                               *
                814.    C *                                                         *
                815.    C *  Common Variables : CYLX,CYLY,LAYRX,LAYRY               *
                816.    C *                                                         *
                817.    C *                                                         *
                818.    C ************************************************************
                819.    C
ISN         1   820.          SUBROUTINE ROTAT1(J,RSPEED,SLOPE,ALPHA)
                821.    C
ISN         2   822.          DOUBLE PRECISION ALPHA(600),CYLX(1202),CYLY(1202),
                823.         +                 LAYRX(30,600),LAYRY(30,600),SLOPE(600)
                824.    C
ISN         3   825.          DOUBLE PRECISION RSPEED
                826.    C
ISN         4   827.          INTEGER          J
                828.    C
ISN         5   829.          PARAMETER        (PI = 3.1415926535897932,HALFPI=PI/2.)
                830.    C
ISN         6   831.          COMMON CYLX,CYLY,LAYRX,LAYRY
                832.    C
                833.    C  Calculate slopes
                834.    C
ISN         7   835.          DO 590 I=2,599
ISN         8   836.             IF (LAYRY(J-1,I+1) .EQ. LAYRY(J-1,I-1)) THEN
ISN         9   837.                SLOPE(I) = 1.E10
ISN        10   838.             ELSE
ISN        11   839.                SLOPE(I) = (LAYRX(J-1,I+1)-LAYRX(J-1,I-1)) / DABS
                840.         +                 (LAYRY(J-1,I+1)-LAYRY(J-1,I-1))
ISN        12   841.             ENDIF
                842.    C
                843.    C  Smooth over any discontinuities in slope caused
                844.    C  by accreting over the previous layer's accretion
                845.    C  limit.
                846.    C
ISN        13   847.             IF (LAYRY(J-1,I-1) .GT. 0.0) THEN
ISN        14   848.                IF (LAYRY(J-1,I+1) .LT. LAYRY(J-1,I-1)) THEN
ISN        15   849.                   IF (SLOPE(I) .LT. SLOPE(I-1)) SLOPE(I) = SLOPE(I-1)
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       17    850.                ENDIF
ISN       18    851.              ENDIF
                852.    C
                853.    C  Calculate position angles.
                854.    C
ISN       19    855.              IF (LAYRX(J-1,I) .EQ. 1.0) THEN
ISN       20    856.                 ALPHA(I) = DABS(HALFPI - ((J-2) * RSPEED))
ISN       21    857.              ELSE IF (LAYRX(J-1,I) .GT. 1.0) THEN
ISN       22    858.                 ALPHA(I) = HALFPI - ((J-2)*RSPEED) +
                859.         +                  DATAN((LAYRX(J-1,I) - 1.) / LAYRY(J-1,I))
ISN       23    860.              ELSE
ISN       24    861.                 IF (DABS(LAYRY(J-1,I)) .LT. 1.0D-10) LAYRY(J-1,I) = 0.0
ISN       26    862.                 IF (DABS(LAYRX(J-1,I)) .LT. 1.0D-10) LAYRX(J-1,I) = 0.0
ISN       28    863.                 ALPHA(I) = DABS(DATAN(LAYRY(J-1,I) /
                864.         +                  (1.0 - LAYRX(J-1,I))) - ((J-2)*RSPEED))
ISN       29    865.              ENDIF
ISN       30    866.      590   CONTINUE
                867.    C
ISN       31    868.            RETURN
ISN       32    869.            END

*STATISTICS*   SOURCE STATEMENTS = 29, PROGRAM SIZE = 1998 BYTES, PROGRAM NAME = ROTAT1    PAGE:   18.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 5 ******
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:31      NAME: MAIN    PAGE:   20

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED          NOTEST   SEQ
                      OPTIMIZE(0)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*

                 870.     C ***************************************************************
                 871.     C * SUBROUTINE ROTAT2 - shifts array indices of current layer*
                 872.     C *                     by an amount corresponding to the     *
                 873.     C *                     rotation increment. Points from the   *
                 874.     C *                     cylinder array must be added to one   *
                 875.     C *                     end.                                  *
                 876.     C *                                                           *
                 877.     C * Input variables :  as defined in main program;           *
                 878.     C *                    J,ROTN                                 *
                 879.     C *                                                           *
                 880.     C * Common variables : CYLX,CYLY,LAYRX,LAYRY                  *
                 881.     C *                                                           *
                 882.     C * Other variables :  K,M - index limits for shifting        *
                 883.     C *                                                           *
                 884.     C ***************************************************************
                 885.     C
ISN         1    886.           SUBROUTINE ROTAT2(J,ROTN)
                 887.     C
ISN         2    888.           DOUBLE PRECISION CYLX(1202),CYLY(1202),LAYRX(30,600),
                 889.          +                 LAYRY(30,600)
                 890.     C
ISN         3    891.           INTEGER J,ROTN
                 892.     C
ISN         4    893.           COMMON CYLX,CYLY,LAYRX,LAYRY
                 894.     C
                 895.     C  Renumber layer array to face new direction of air stream,
                 896.     C  dropping increment number of points at the 'front' end.
                 897.     C
ISN         5    898.           K = 600-ROTN
ISN         6    899.           M = 600-ROTN+1
ISN         7    900.           DO 580 I = 1,K
ISN         8    901.              LAYRX(J,I) = LAYRX(J,I+ROTN)
ISN         9    902.              LAYRY(J,I) = LAYRY(J,I+ROTN)
ISN        10    903.        580  CONTINUE
                 904.     C
                 905.     C  Add increment number of points from the bare cylinder to
                 906.     C  the 'back' end.
                 907.     C
ISN        11    908.           DO 585 I = M,600
ISN        12    909.              LAYRX(J,I) = CYLX(299 + (J-1)*ROTN + I)
ISN        13    910.              LAYRY(J,I) = CYLY(299 + (J-1)*ROTN + I)
ISN        14    911.        585  CONTINUE
ISN        15    912.           RETURN
ISN        16    913.           END

*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 780 BYTES, PROGRAM NAME = ROTAT2    PAGE:   20.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 6 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                914.    C
                915.    C ***********************************************************
                916.    C *  Function SPCBET - Estimates Beta(alpha) for a         *
                917.    C *           simulated droplet size distribution, for an  *
                918.    C *           mvd either smaller or larger than 20 microns.*
                919.    C *           The polynomial function is scaled to (alpha  *
                920.    C *           max + tail length), where the tail length    *
                921.    C *           is a function of the mvd.                    *
                922.    C *                                                        *
                923.    C *  Input Variables : as defined for Main program;       *
                924.    C *                     ALPHA,ALPHM,BZERO,SMDROP,TAILEN    *
                925.    C *                                                        *
                926.    C *  Other Variables : X - scaled angle = Alpha /          *
                927.    C *                       (Alphm + Tailen)                 *
                928.    C *                                                        *
                929.    C ***********************************************************
                930.    C
ISN        1    931.          DOUBLE PRECISION FUNCTION SPCBET(ALPHA,SMDROP,BZERO,
                932.         +                                 ALPHM,TAILEN)
                933.    C
ISN        2    934.          DOUBLE PRECISION ALPHA,ALPHM,BZERO,TAILEN,X
                935.    C
ISN        3    936.          LOGICAL          SMDROP
                937.    C
ISN        4    938.          IF (ALPHA .GE. (ALPHM+TAILEN)) THEN
ISN        5    939.            SPCBET = 0.0
ISN        6    940.          ELSE
ISN        7    941.            X = ALPHA / (ALPHM+TAILEN)
ISN        8    942.            IF (SMDROP) THEN
ISN        9    943.              SPCBET = BZERO * (1.0 + 0.2195 * X - 3.084 *
                944.         +                (X ** 2) + 1.866 * (X ** 3))
ISN       10    945.            ELSE
ISN       11    946.              SPCBET = BZERO * (1.0 + 0.0287 * X - 1.936 *
                947.         +                (X ** 2) + 2.484 * (X ** 3) - 4.112
                948.         +                * (X ** 4) + 2.538 * (X ** 5))
ISN       12    949.            ENDIF
ISN       13    950.            IF (SPCBET .LT. 0.0) SPCBET = 0.0
ISN       15    951.          ENDIF
ISN       16    952.          RETURN
ISN       17    953.          END
```

*STATISTICS* SOURCE STATEMENTS = 16, PROGRAM SIZE = 708 BYTES, PROGRAM NAME = SPCBET PAGE: 21.

*STATISTICS* NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 7 ******

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:32     NAME: MAIN    PAGE:   22

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED          NOTEST   SEQ
                      OPTIMIZE(0)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                          *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*

                   954.    C
                   955.    C ************************************************************
                   956.    C * SUBROUTINE WTBET - Calculates the average local          *
                   957.    C *                  collision efficency according to      *
                   958.    C *                  the input volume weighted droplet     *
                   959.    C *                  size spectrum.                        *
                   960.    C *                                                        *
                   961.    C * Input Variables : as defined for Main program;          *
                   962.    C *                  ALPHA,AM,BZ,SMDROP,WGHT               *
                   963.    C *                                                        *
                   964.    C * Output Varibles : as defined for Main program;         *
                   965.    C *                  BETA                                  *
                   966.    C *                                                        *
                   967.    C ************************************************************
                   968.    C
ISN        1       969.          SUBROUTINE WTBET(ALPHA,SMDROP,BZ,AM,WGHT,BETA)
                   970.    C
ISN        2       971.          DOUBLE PRECISION AM(18),BZ(18),WGHT(18)
                   972.    C
ISN        3       973.          DOUBLE PRECISION ALPHA,BETA
                   974.    C
ISN        4       975.          LOGICAL          SMDROP
                   976.    C
ISN        5       977.                BETA = 0.0
ISN        6       978.                DO 22 L=1,18
ISN        7       979.                   IF (WGHT(L) .NE. 0.0) THEN
ISN        8       980.                      BETA = BETA + WGHT(L) * MONBET(
                   981.           +                    ALPHA(I),BZ(L),AM(L),SMDROP)
ISN        9       982.                   ENDIF
ISN       10       983.             22    CONTINUE
ISN       11       984.          RETURN
ISN       12       985.          END

*STATISTICS*   SOURCE STATEMENTS = 12, PROGRAM SIZE = 654 BYTES, PROGRAM NAME = WTBET     PAGE:   22.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 8 ******
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:32      NAME: MAIN    PAGE:   23
SUMMARY OF MESSAGES AND STATISTICS FOR ALL COMPILATIONS
*STATISTICS*   SOURCE STATEMENTS = 161, PROGRAM SIZE = 27174 BYTES, PROGRAM NAME = MAIN    PAGE:    1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
*STATISTICS*   SOURCE STATEMENTS = 64, PROGRAM SIZE = 12132 BYTES, PROGRAM NAME = DRAW    PAGE:   12.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 2 ******
*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 586 BYTES, PROGRAM NAME = MONBET    PAGE:   15.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 3 ******
*STATISTICS*   SOURCE STATEMENTS = 27, PROGRAM SIZE = 1362 BYTES, PROGRAM NAME = PARAM    PAGE:   16.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 4 ******
*STATISTICS*   SOURCE STATEMENTS = 29, PROGRAM SIZE = 1998 BYTES, PROGRAM NAME = ROTAT1    PAGE:   18.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 5 ******
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 780 BYTES, PROGRAM NAME = ROTAT2    PAGE:   20.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 6 ******
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 708 BYTES, PROGRAM NAME = SPCBET    PAGE:   21.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 7 ******
*STATISTICS*   SOURCE STATEMENTS = 12, PROGRAM SIZE = 654 BYTES, PROGRAM NAME = WTBET    PAGE:   22.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 8 ******
******* SUMMARY STATISTICS ******* 0 DIAGNOSTICS GENERATED. HIGHEST SEVERITY CODE IS 0.
```

# Appendix III

## Approximation of the mvd from Stagnation Line Thickness

The measurement of water droplet sizes and size distributions is one of the major experimental problems in icing studies. The very common measurement technique, known as the oil slide method, has been shown by Makkonen and Stallabrass (1984) to overestimate the median volume diameters of observed droplet spectra in comparison to measurements made with an optical sizing probe, the Forward Scattering Spectrometer Probe, or FSSP, manufactured by Particle Measuring Systems and described by Knollenberg (1981). Makkonen and Stallabrass provide an empirical correction formula for oil slide measured mvds in the range 10 to 40 μm, as follows:

$$mvd_{FSSP} = 8.8 + 0.27\ mvd_{oil\ slide} \qquad A.1$$

All of the wind tunnel experiments performed in the FROST tunnel and described in this thesis have had droplet size distributions determined by the oil slide method. The mvds derived from this data appear to be too large, not only in light of Makkonen and Stallabrass' results, but also because the theoretical collection efficiencies calculated

for these mvds imply a minimum stagnation line thickness much larger than that observed.

In order to supply the most accurate input data possible to the model, these oil slide measurement should be corrected. However, some of the circular cylinder cases lie outside of the range given for Equation A.1. For these cases an approximate mvd has been derived from the observed stagnation line thicknesses of accreted ice, using the method outlined below, similar to the rotating cylinder methods which have been used for the same purpose.

For cases which lie inside the range of validity, these approximate mvds agree well (within about 10 per cent) with the results of Equation A.1. The correlation of Makkonen and Stallabrass has therefore been used for all cases within the appropriate range.

Outside of that range, however, it is important to remember that the approximation given here is derived independently of the actual size spectrum, and represents a droplet size corresponding to the average value of $\beta_0$ for the time period of the accretion layer. It has been used here only because of the absence of a reliable calibration for oil slide measurements of droplets larger than 40 μm.

The method is as follows:

1. The stagnation line thickness (Th) of the chosen layer is measured from a photograph of the accretion in cross-section.
2. A first estimate of $\beta_0$ is calculated from applying

the dry growth equation to the stagnation line:

$$\beta_0 = (\rho_0 \text{ Th}) / (\text{U lwc } \tau) \qquad \text{A.2}$$

A best estimate of $\rho_0$ may be made as in Chapter IV. For lwc, the first guess is made using the oil slide mvd, and the measured rate of mass collected on a small rotating cylinder, and calculated by the algorithm given by Stallabrass (1978).

3. Calculate the Stokes parameter,

$$\phi = 9 \rho_a^2 D_c U / (\mu \rho_d), \qquad \text{A.3}$$

then read from Figure II.3, a value of K corresponding to $\phi$ and $\beta_0$.

4. Derive a new value of mvd from the definition of K:

$$K = (\rho_d / 18 \mu) (\text{mvd}^2 U / D_c) \qquad \text{A.4}$$

5. Recalculate lwc using the new mvd.

6. Recalculate $\beta_0$ from new lwc. Repeat steps 2 through 6 until the values of $\beta_0$, mvd and lwc appear to have converged.

For the examples given below, the rotating cylinder diameter was 2.46 mm.

*Example 1:*

oil slide mvd = 24.2 μm

Th = 1.49 mm

U = 10.7 m/sec

$\tau$ = 600 sec

$\rho_0$ = 750 kg/m$^3$

rotating cylinder mass = 0.029 g/min

revised mvd = 13.8 μm

mvd from Equation A.1 = 15.3 μm

*Example 2:*

oil slide mvd = 136 μm

Th = 13.5 mm

U = 10.24 m/sec

$\tau$ = 1800 sec

$\rho_0$ = 890 kg/m$^3$

rotating cylinder mass = 0.122 g/min

revised mvd = 62 μm

## References

Knollenberg, R.G. 1981: Techniques for Probing Cloud Microstructure. Clouds, Their Formation, Optical Properties, and Effects, ed.s P.V. Hobbs and A. Deepak. Academic Press, 15-92.

Makkonen, L. and Stallabrass, J.R. 1984: Ice Accretion on Cylinders and Wires. National Research Council, Division of Mechanical Engineering, Technical Report TR-LT-005, 50 pp.

Stallabrass, J.R. 1978: National Research Council, Division of Mechanical Engineering, Technical Report TR-LT-92, 26 pp., 7 fig.s, + 2 App.s.

## Appendix IV

### A Geometrical Derivation of Equation V.3

Figure A.1 illustrates two droplet trajectories impinging on an airfoil, at zero angle of attack. If the initial separation of the trajectories is $\Delta y$, then the distance separating them (in the direction normal to the lower trajectory) will have increased at the point of impact to, say, $(\Delta y + a)$. Referring to Figure A.1, if the angle $\gamma$ is the departure from horizontal of the tangent to the upper trajectory at the point X, it follows that for small $\Delta \ell$:

$$\cos(\alpha + \gamma) \simeq (\Delta y + a) / \Delta \ell = \beta + (a / \Delta \ell), \qquad A.5$$

using the definition of $\beta$ given in Chapter II.

For airfoils, $\gamma$ is always $<< \alpha$, so:

$$\cos(\alpha + \gamma) \simeq \cos \alpha \qquad A.6$$

Now if it is assumed that $(a / \Delta \ell)$ is approximately a constant, and applying the boundary condition at $\alpha = 0$, this gives:

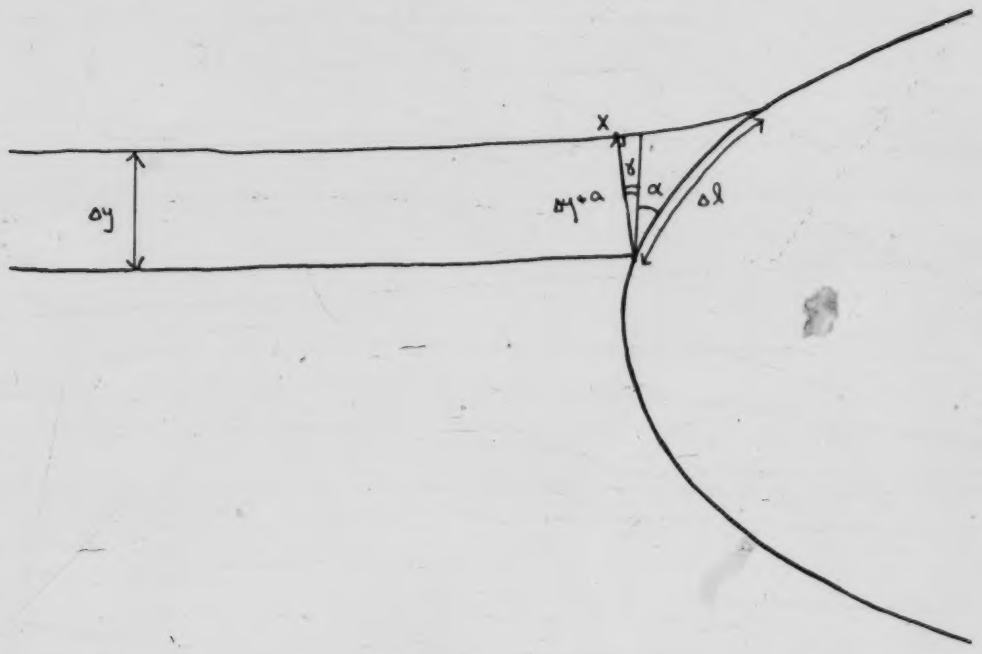



Figure A.1 - Droplet trajectories impinging on an airfoil.

$$\beta \simeq \cos \alpha - (1 - \beta_0), \qquad \text{A.7}$$

which is Equation V.3. The above approximation should then hold when the impinging trajectories have small curvature.

# Appendix V

## Program Listing: OMNIFOIL

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
 1.   C        ********** Airfoil Rime Icing Model **********
 2.   C
 2.2  C        Karen J. Finstad        June, 1986
 2.4  C
 3.   C        This program will calculate and plot rime ice accretion
 4.   C     layers on an airfoil, as well as calculate the mass and
 5.   C     mean density of each layer. Local collision efficiencies
 6.   C     for each point on the surface are calculated as a cosine
 7.   C     function of the current local surface angle
 8.   C     and scaled to the value of stagnation line collision
 9.   C     efficiency (Beta zero), which is determined from droplet
10.   C     trajectory calculations.
11.   C
12.   C        Beta zero is updated on every fourth layer to account
13.   C     for the changing flow around the accretion. Layer time
14.   C     steps should be chosen so that Beta zero is updated at
15.   C     reasonable intervals, say after a maximum thickness of no
16.   C     more than 1/10 of the chord length has been accreted.
17.   C
18.   C        The maximum accretion angle, alpha max, is determined
19.   C     from Beta zero.
20.   C
21.   C        When the angle of attack is zero, a "simulated spectrum"
22.   C     version of local collision efficiency distribution may be
23.   C     used to approximate the effects of droplet size spectra
24.   C     with median volume diameters less than about 20 microns.
25.   C     Larger drops are well represented by the monodisperse
26.   C     version of collision efficiency parameterisation at the mvd.
27.   C
28.   C        For other angles of attack, an approximate "skewed"
29.   C     distribution is used, which should apply to all droplet
30.   C     sizes.
31.   C
32.   C        Local density is approximated by a polynomial
33.   C     function of the local surface slope.
34.   C
35.   C        The output consists of a plot description file written into
36.   C     -pdf attached to unit 8, and a mass and density table written
37.   C     into -mass attached to unit 9. (A negative sign identifies
38.   C     temporary files on MTS.)
39.   C
40.   C        The plot description file is produced using the system
41.   C     subroutine library *PLOTLIB. All of the plotting is done
42.   C     in a single program subroutine, so that other available
43.   C     plotting routines may be substituted if desired.
44.   C
45.   C        An additional external subroutine , from the *IMSLDPLIB
46.   C     library, is used in the solution of the potential flow.
47.   C     Other matrix inversion routines may be substituted.
48.   C
49.   C        Please note: in order to run *PLOTLIB subroutines (or any
50.   C     subroutines which have not been compiled in Fortran77 and
51.   C     which contain character strings or variables in their
```

```
*....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      C    argument list), in conjunction with this program, (or any
      C    Fortran77 program) the command @PROCESS(list of subroutines)
      C    MUST appear in ROW 1, COLUMN 1 if the subroutines are
      C    called directly from MN. If they are called from
      C    another subroutine, the @PROCESS command must be placed
      C    in COLUMN 1 OF THE LINE IMMEDIATELY FOLLOWING THE END
      C    STATEMENT OF THE PREVIOUS ROUTINE.
      C
      C        To compile (on MTS), first empty -mass and -pdf. Then
      C    $RUN *FORTRANVS SCARDS=OMNIFOIL SPUNCH=FOILRUN
      C
      C        To run, enter $RUN FOILRUN+*IMSLDPLIB+*PLOTLIB.
      C    You will be prompted for the following input parameters:
      C
      C        median volume droplet DIAMETER in microns
      C
      C        y or n to simulated spectrum option
      C
      C        airfoil chord length in metres
      C
      C        NACA airfoil height as fraction of chord
      C
      C        free stream air speed in metres per second
      C
      C        liquid water content in kg per cubic metre
      C
      C        time step per layer in seconds
      C
      C        total number of layers
      C
      C        angle of attack of the free stream in degrees
      C
      C        OR, input data may be entered in a file. You must
      C    then specify "5 = filename" in the $RUN command.
      C
      C    ********************** Subprograms **********************
      C
      C
      C    INTERNAL:
      C
      C        SUBROUTINE DRAW - Plots airfoil profile and all
      C     accreted ice layers with appropriate legend.
      C
      C        SUBROUTINE DROPS - Integrates droplet trajectories
      C     and calculates stagnation line collision efficiency.
      C     Subroutines called only from within DROPS (COLLID,
      C     FLOWXY,KCALC,POFLO,TRAJEC) are described in the program
      C     summary to DROPS.
      C
      C        DOUBLE PRECISION FUNCTION ONEBET - Estimates local
      C     collision efficiency as a function of surface slope for
      C     a monodisperse droplet population at the mvd.
      C
      C        DOUBLE PRECISION FUNCTION PHIBET - calculates local
      C     collision efficiency as a function of slope w.r.t to
      C     free stream direction, and distance from the maximum
```

```
C     local value of collision efficiency (set to 0.95 times
C     the value for zero angle of attack). The location of
C     the maximum is estimated from a correlation derived
C     from previous numerical results .
C
C
C        DOUBLE PRECISION FUNCTION SPCBET - Estimates local
C     collision efficiency as a function of surface slope
C     for an average distribution shape of droplet sizes,
C     scaled to the mvd. The function assumes that small
C     droplets dominate in the spectrum, with mvd less than
C     about 20 microns.
C
C
C
C     EXTERNAL:
C
C        *IMSLDPLIB - The subroutine LEQT1F is used to solve
C     the matrix of potential vorticites for the potential
C     flow calculation in subroutine DROPS.
C
C        *PLOTLIB  -  The subroutines PLOTS, METRIC, PLOT, AXIS2,
C     LINE, SYMBOL and NUMBER (called from SUBROUTINE DRAW) are
C     found in the public system subroutine library *PLOTLIB.
C     (See note above regarding external subroutines in Fortran77).
C
C     ****************** Variable Dictionary ********************
C
C       ALPHA        - array of angles between surface normal
C                      and the free stream direction
C
C       ALPHM        - Alpha max, the maximum impingement surface
C                      angle
C
C       APHI         - parameter for determining the distribution
C                      of collision efficiency and of density
C                      about the position of maximum c.e. when
C                      phi is non-zero.
C
C       AREA         - cross sectional area between
C                      adjacent points in the current layer
C                      (dimensioned in metres)
C
C       AZERO        - the surface angle of maximum collision
C                      efficiency when phi is non-zero,
C                      determined from an approximating
C                      expression.
C
C       BETA         - local collision efficiency
C
C       BZERO        - stagnation line value of the local collision
C                      efficiency
C
C       CHORD        - length of airfoil chord plus accreted ice
C
C       CHORD1       - length of bare airfoil chord in metres
```

```
*....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
C                    (INPUT)
C
C     DIAG           - cross diagonal of the quadrilateral
C                      of AREA
C
C     DROPDI         - droplet diameter in microns (INPUT)
C
C     H              - airfoil thickness as a fraction of the
C                      chord length (INPUT)
C
C     HALFPI         - pi/2 (Fortran77 constant PARAMETER)
C
C     LAREA          - cross sectional area of the current layer
C                      in sq. metres
C
C     LAYERS         - integer number of layers
C
C     LAYRX,LAYRY    - array containing non - dimensional x and y
C                      coordinates of the accretion surfaces for
C                      up to 30 accretion layers
C
C     LDENS          - mean density of the current layer in kg/cubic m
C
C     LMASS          - mass of the current layer in kg per unit length
C
C     LWC            - liquid water content in kg per cubic metre
C                      (INPUT)
C
C     L1,L2,L3       - dimensioned sides of the quadrilateral area
C     L4               defined by two adjacent points on the
C                      current surface and two on the surface below
C
C     MASS           - array of masses per unit length for areas
C                      defined by adjacent points, in kg
C
C     NPOINT         - number of points defining the initial
C                      airfoil surface (note not the same as
C                      for potential flow calculation)
C
C     PFOILX,        - arrays of non-dimensional x,y points
C     PFOILY           of the front of the initial surface,
C                      for plotting
C
C     PHI            - angle of attack of the free stream
C                      (INPUT)
C
C     PI             - pi (Fortran77 constant PARAMETER)
C
C     RHO            - array of local densities in kg/cubic metre
C
C     SIMSPC         - logical variable, true if simulated spectrum
C                      option is chosen
C
C     SIMOPT         - character variable for reading in response to
C                      simulated spectrum option query (INPUT)
C
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                220.    C       SLOPE        - array of local surface slopes calculated
                221.    C                      at each surface point
                222.    C
                223.    C       S1,S2        - quantities used to determine the quadrilaterals
                224.    C                      for the layer area calculation
                225.    C
                226.    C       TAILEN       - filter length for optional smoothing filter
                227.    C                      applied to c. e. curve
                228.    C
                229.    C       TAU          - accretion time step in seconds (INPUT)
                230.    C
                231.    C       TH           - accumulated stagnation line thickness
                232.    C
                233.    C       THICK        - array of ice thicknesses normal to surface
                234.    C
                235.    C       TMASS        - total mass of the accretion
                236.    C
                237.    C       VEL          - free stream air speed in m per sec (INPUT)
                238.    C
                239.    C       XFOIL,       - arrays containing  dimensional x and
                240.    C       YFOIL          y coordinates of the airfoil surface
                241.    C
                242.    C
                243.    C ************************************************************
                244.    C
ISN        1    245.            DOUBLE PRECISION ALPHA(600),GAMMA(41),LAYRX(20,300),
                246.           +                 LAYRY(20,300),PFOILX(302),PFOILY(302),
                247.           +                 RHO(300),SLOPE(300),THICK(300),
                248.           +                 XFOIL(600),YFOIL(600)
                249.    C
ISN        2    250.            DOUBLE PRECISION AZERO,ALPHM,APHI,AREA,BETA,BZERO,
                251.           +                 CHORD,CHORD1,DIAG,DROPDI,H,LAREA,LDENS,
                252.           +                 LMASS,LWC,L1,L2,L3,L4,MASS,PHI,S1,
                253.           +                 S2,TAILEN,TAU,TH,TMASS,VEL
                254.    C
ISN        3    255.            INTEGER          LAYERS,NPOINT
                256.    C
ISN        4    257.            CHARACTER*1      SIMOPT
                258.    C
ISN        5    259.            LOGICAL          SIMSPC
                260.    C
ISN        6    261.            PARAMETER (PI = 3.1415926535897932,HALFPI = PI/2)
                262.    C
ISN        7    263.            COMMON /MN/XFOIL,YFOIL,GAMMA/DR/LAYRX,LAYRY,PFOILX,PFOILY
                264.    C
                265.    C  Prepare mass table and plot description files.
                266.    C
ISN        8    267.            OPEN (UNIT = 8, FILE = '-MASS', STATUS = 'UNKNOWN')
ISN        9    268.            OPEN (UNIT = 9, FILE = '-PDF', STATUS = 'UNKNOWN')
                269.    C
                270.    C  Set up headings for mass and density table in -MASS.
                271.    C
ISN       10    272.            WRITE (8,110)
ISN       11    273.        110 FORMAT (///,15X,'Mass of Ice Accreted',//,5X,'Layer no.',
                274.           +        7X,'Mass',7X,'Mean Density',//,
                275.           +        20X,'(kg/m)',9X,'(kg/cu.m)',//,
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                      276.          +         ' **********************************************')
                      277.     C
                      278.     C   Write program title and intro to screen.
                      279.     C
ISN         12        280.           WRITE (6,120)
ISN         13        281.       120 FORMAT (/////,' ************  AIRFOIL RIME ICING MODEL  *****',
                      282.          +        '************',//,17X,' Version Mar. 12, 1986',//,
                      283.          +        20X,' K. J. Finstad',/,17X,
                      284.          +        ' Meteorology Division',/,12X,' The University of Alberta'
                      285.          +        ,//,' Please enter parameters to begin accretion: ',//)
                      286.     C
                      287.     C    Prompt user for input parameters.
                      288.     C
ISN         14        289.           WRITE (6,*) 'Median Volume Droplet DIAMETER in microns? '
ISN         15        290.           READ  (5,*) DROPDI
                      291.     C
                      292.     C    Convert to metres
                      293.     C
ISN         16        294.           DROPDI = DROPDI*1.0E-6
                      295.     C
                      296.     C   Set flags for requested treatment of droplet spectrum.
                      297.     C
ISN         17        298.                WRITE (6,*)
ISN         18        299.                WRITE (6,*) 'Do you wish simulated spectrum weighting? '
ISN         19        300.                READ  (5,8) SIMOPT
ISN         20        301.           8    FORMAT (1A1)
                      302.     C
                      303.     C  For simulated spectrum, calculate 'tail' length from
                      304.     C  median volume diameter.
                      305.     C
ISN         21        306.           IF ((SIMOPT .EQ. 'Y') .OR. (SIMOPT .EQ. 'y')) THEN
ISN         22        307.             SIMSPC = .TRUE.
ISN         23        308.              TAILEN = 0.13 * DEXP(-DROPDI / 7.0E-5)
ISN         24        309.           ELSE
ISN         25        310.             SIMSPC = .FALSE.
ISN         26        311.             TAILEN = 0.0
ISN         27        312.           ENDIF
                      313.     C
ISN         28        314.           WRITE (6,*)
ISN         29        315.           WRITE (6,*) 'Airfoil chord length in metres? '
ISN         30        316.           READ  (5,*) CHORD1
                      317.     C
                      318.     C  Set current chord length equal to bare airfoil
                      319.     C
ISN         31        320.           CHORD = CHORD1
                      321.     C
ISN         32        322.           WRITE (6,*)
ISN         33        323.           WRITE (6,*) 'Airfoil height as fraction of chord? '
ISN         34        324.           READ  (5,*) H
                      325.     C
ISN         35        326.           WRITE (6,*)
ISN         36        327.           WRITE (6,*) 'Free stream speed in m / sec? '
ISN         37        328.           READ  (5,*) VEL
                      329.     C
ISN         38        330.           WRITE (6,*)
ISN         39        331.           WRITE (6,*) 'Liquid water content in kg / cubic metre? '
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN      40   332.          READ  (5,*) LWC
              333.    C
ISN      41   334.          WRITE (6,*)
ISN      42   335.          WRITE (6,*) 'Time step per layer in seconds? '
ISN      43   336.          READ  (5,*)  TAU
              337.    C
ISN      44   338.          WRITE (6,*)
ISN      45   339.          WRITE (6,*) 'Total number of layers? '
ISN      46   340.          READ  (5,*) LAYERS
              341.    C
              342.    C   Bare surface will be layer 1, so add 1 to LAYERS
              343.    C
ISN      47   344.          LAYERS = LAYERS + 1
              345.    C
ISN      48   346.          WRITE (6,*)
ISN      49   347.          WRITE (6,*) 'Angle of attack in degrees? '
ISN      50   348.          READ  (5,*) PHI
              349.    C
              350.    C  Convert angle of attack to radians, and calculate
              351.    C  surface angle of the max collision efficiency point
              352.    C  for non-zero attack angles (AZERO).
              353.    C
ISN      51   354.          AZERO = DLOG10(PHI + 1.) / (0.001 * DEXP(30.5 * H))
ISN      52   355.          PHI   = (PHI * 2.0 * PI) / 360.0
ISN      53   356.          AZERO = (AZERO * 2.0 * PI) / 360.0
              357.    C
              358.    C  Fill array of points defining the initial  surface of the
              359.    C  airfoil in dimensional coordinates. The complete
              360.    C  surface contains NPOINT points.
              361.    C
ISN      54   362.           NPOINT = 600
              363.    C
ISN      55   364.          DO 100 I = 1,NPOINT
ISN      56   365.             XFOIL(I) = 0.5 * (1.0 - DCOS(2.0 * PI * DBLE(I-1)/
              366.         +                DBLE(NPOINT)))
ISN      57   367.             YFOIL(I) = 5.0 * H * (0.2969 * DSQRT(XFOIL(I)) -
              368.         +                0.126 * XFOIL(I) - 0.3516 * XFOIL(I) ** 2
              369.         +                + 0.2843 * XFOIL(I) ** 3 -
              370.         +                0.1015 * XFOIL(I) ** 4) * CHORD1
ISN      58   371.             IF (I .GT. 301) YFOIL(I) = -YFOIL(I)
ISN      60   372.             XFOIL(I) = XFOIL(I) * CHORD1
ISN      61   373.          100  CONTINUE
              374.    C
              375.    C  Fill arrays of non-dimensional coordinates for later plotting.
              376.    C
ISN      62   377.          DO 101 I=1,150
ISN      63   378.             PFOILX(150+I) = XFOIL(I) / CHORD1
ISN      64   379.             PFOILY(150+I) = YFOIL(I) / CHORD1
ISN      65   380.             PFOILX(I)     = XFOIL(450+I) / CHORD1
ISN      66   381.             PFOILY(I)     = YFOIL(450+I) / CHORD1
ISN      67   382.          101  CONTINUE
              383.    C
              384.    C  Copy front part of  surface to array which will contain
              385.    C  all of the layers.
              386.    C
ISN      68   387.          DO 200 I = 1,NPOINT / 4
```

```
                                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN      69      388.                LAYRX(1,I) = XFOIL((3 * NPOINT / 4) + I)
ISN      70      389.                LAYRY(1,I) = YFOIL((3 * NPOINT / 4) + I)
ISN      71      390.          200   CONTINUE
ISN      72      391.                DO 210 I = (NPOINT / 4) + 1, NPOINT / 2
ISN      73      392.                  LAYRX(1,I) = XFOIL(I - (NPOINT / 4))
ISN      74      393.                  LAYRY(1,I) = YFOIL(I - (NPOINT / 4))
ISN      75      394.          210   CONTINUE
                 395.          C
                 396.          C  Call Subroutine DROPS to calculate Beta zero. Alpha max is
                 397.          C  then determined from Beta zero.
                 398.          C
ISN      76      399.                CALL DROPS(DROPDI,VEL,CHORD1,NPOINT,BZERO)
ISN      77      400.                ALPHM = DACOS(1.0 - BZERO)
                 401.          C
                 402.          C
                 403.          C  Initialize total mass and begin loop to accrete ice layers.
                 404.          C
ISN      78      405.                TMASS = 0.0
ISN      79      406.                DO 300 J = 2,LAYERS
                 407.          C
                 408.          C  Check for recalculation of BZERO every fifth layer
                 409.          C
ISN      80      410.                IF ((J .EQ. 6) .OR. (J .EQ. 11) .OR. (J .EQ. 16)) THEN
                 411.          C
                 412.          C  Copy new shape to arrays for potential flow calculation in DROPS
                 413.          C
ISN      81      414.                   DO 220 I=1,NPOINT / 4
ISN      82      415.                      XFOIL(I + (3*NPOINT/4)) = LAYRX(J-1,I)
ISN      83      416.                      YFOIL(I + (3*NPOINT/4)) = LAYRY(J-1,I)
ISN      84      417.           220        CONTINUE
ISN      85      418.                   DO 230 I = (NPOINT / 4) + 1,NPOINT / 2
ISN      86      419.                      XFOIL(I - NPOINT / 4) = LAYRX(J-1,I)
ISN      87      420.                      YFOIL(I - NPOINT / 4) = LAYRY(J-1,I)
ISN      88      421.           230        CONTINUE
                 422.          C
                 423.          C  Update length of airfoil chord plus accreted ice
                 424.          C
ISN      89      425.                CHORD = CHORD1 + (((J-2) * BZERO * VEL * LWC * TAU)/890.0)
                 426.          C
                 427.          C  Update BZERO, and ALPHM if necessary
                 428.          C
ISN      90      429.                   CALL DROPS(DROPDI,VEL,CHORD,NPOINT,BZERO)
ISN      91      430.                   IF (DACOS(1.0 - BZERO) .LT. ALPHM) ALPHM =
                 431.               +        DACOS(1.0 - BZERO)
ISN      93      432.                  ENDIF
                 433.          C
                 434.          C  Calculate slope and surface angle from finite differences for
                 435.          C  each point on current surface.
                 436.          C
ISN      94      437.                DO 400 I = 2,299
ISN      95      438.                   SLOPE(I) = (LAYRX(J-1,I+1) - LAYRX(J-1,I-1)) /
                 439.               +              DABS(LAYRY(J-1,I+1) - LAYRY(J-1,I-1))
ISN      96      440.                   IF (LAYRY(J-1,I+1) .EQ. LAYRY(J-1,I-1))
                 441.               +        SLOPE(I) = 1.0E10
ISN      98      442.                   IF (LAYRY(J-1,I+1) .LT. LAYRY(J-1,I-1)) THEN
ISN      99      443.                      ALPHA(I) = PI - DATAN(SLOPE(I))
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN     100   444.            ELSE
ISN     101   445.               ALPHA(I) = DATAN(SLOPE(I))
ISN     102   446.            ENDIF
ISN     103   447.      400  CONTINUE
              448.      C
              449.      C  Calculate for each point local density, collision efficiency and
              450.      C  ice thickness. The method of calculation for Beta and rho
              451.      C  depend on the values of SIMSPC and PHI.
              452.      C
ISN     104   453.            DO 500 I = 2,299
ISN     105   454.              IF (PHI .EQ. 0.0) THEN
ISN     106   455.                  RHO(I) =890. *
              456.           +              (1.0 - 0.163 * DABS(ALPHA(I)/(ALPHM+TAILEN)) +
              457.           +               0.235 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 2) -
              458.           +               2.357 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 3) +
              459.           +               1.374 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 4) +
              460.           +               0.0188 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 5))
ISN     107   461.               ELSE
ISN     108   462.                   APHI   = ((-ALPHA(I) - AZERO) / (HALFPI - AZERO)
              463.           +             * HALFPI)
ISN     109   464.                   RHO(I) = 890.0 *
              465.           +                (1.0 - (0.163 * DABS(APHI/HALFPI)) +
              466.           +                (0.235 * DABS(APHI/HALFPI) ** 2) -
              467.           +                (2.357 * DABS(APHI/HALFPI) ** 3) +
              468.           +                (1.374 * DABS(APHI/HALFPI) ** 4) +
              469.           +                (0.0188 * DABS(APHI/HALFPI) ** 5))
ISN     110   470.               ENDIF
              471.      C        IF (ALPHA(I) .GT. HALFPI) RHO(I) =170.
              472.      C
              473.      C  Make sure density does not become too small
              474.      C
ISN     111   475.               IF (RHO(I) .LT. 170.) RHO(I) =170.
              476.      C
              477.      C  Calculate local collision efficiency
              478.      C
ISN     113   479.               IF (PHI .NE. 0.0) THEN
ISN     114   480.                  BETA = 0.95 * SPCBET(APHI,BZERO,HALFPI,0.0)
ISN     115   481.               ELSE IF (SIMSPC) THEN
ISN     116   482.                  BETA = SPCBET(ALPHA(I),BZERO,ALPHM,TAILEN)
ISN     117   483.               ELSE
ISN     118   484.                  BETA = ONEBET(ALPHA(I),BZERO,ALPHM)
ISN     119   485.               ENDIF
              486.      C
              487.      C  Calculate local ice thickness perpendicular to surface
              488.      C
ISN     120   489.               THICK(I) = BETA * VEL * TAU * LWC / RHO(I)
              490.      C
              491.      C  Define new layer surface coordinates.
              492.      C
ISN     121   493.               LAYRX(J,I) = LAYRX(J-1,I) - (THICK(I) *
              494.           +                           DCOS(DATAN(SLOPE(I))))
ISN     122   495.               LAYRY(J,I) = LAYRY(J-1,I) + (THICK(I) *
              496.           +                           DSIN(DATAN(SLOPE(I))))
ISN     123   497.          500  CONTINUE
ISN     124   498.               LAYRX(J,1)   = LAYRX(J-1,1)
ISN     125   499.               LAYRX(J,300) = LAYRX(J-1,300)
```

```
                                 *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN      126      500.                LAYRY(J,1)   = LAYRY(J-1,1)
ISN      127      501.                LAYRY(J,300) = LAYRY(J-1,300)
                  502.          C
                  503.          C  Calculate mass and mean density of the layer from
                  504.          C  the sum of areas between surface points on successive
                  505.          C  layers. First, initialize layer mass and area.
                  506.          C
ISN      128      507.                LMASS = 0.0
ISN      129      508.                LAREA = 0.0
ISN      130      509.                DO 560 I = 2,298
ISN      131      510.                   IF ((THICK(I) .GT. 0.0).AND.(THICK(I+1) .GT. 0.0)) THEN
ISN      132      511.                      L4 = THICK(I)
ISN      133      512.                      L2 = THICK(I+1)
ISN      134      513.                      L1 = DSQRT((LAYRX(J-1,I+1) - LAYRX(J-1,I))**2 + (LAYRY
                  514.               +             (J-1,I+1) - LAYRY(J-1,I))**2)
ISN      135      515.                      L3 = DSQRT((LAYRX(J,I+1) - LAYRX(J,I))**2 + (LAYRY
                  516.               +             (J,I+1) - LAYRY(J,I))**2)
ISN      136      517.                      DIAG = DSQRT((LAYRX(J,I+1)-LAYRX(J-1,I))**2 + (LAYRY
                  518.               +               (J,I+1) - LAYRY(J-1,I))**2)
ISN      137      519.                      S1   = 0.5 * (DIAG+L1+L2)
ISN      138      520.                      S2   = 0.5 * (DIAG+L3+L4)
ISN      139      521.                      AREA = DSQRT(S1 * (S1-DIAG) * (S1-L1) * (S1-L2)) +
                  522.               +                 DSQRT(S2 * (S2-DIAG) * (S2-L3) * (S2-L4))
ISN      140      523.                      MASS = ((RHO(I) + RHO(I+1)) / 2.0) * AREA
ISN      141      524.                      LMASS   = LMASS + MASS
ISN      142      525.                      LAREA   = LAREA + AREA
ISN      143      526.                   ENDIF
ISN      144      527.             560  CONTINUE
                  528.          C
                  529.          C  Calculate mean density.
                  530.          C
ISN      145      531.                LDENS = LMASS/LAREA
                  532.          C
                  533.          C  Write results to mass table.
                  534.          C
ISN      146      535.                WRITE (8,570) J-1,LMASS,LDENS
ISN      147      536.             570  FORMAT (//,8X,I2,7X,F9.5,7X,F9.5)
                  537.          C
                  538.          C  Add mass of current layer to total mass.
                  539.          C
ISN      148      540.                TMASS = TMASS + LMASS
                  541.          C
                  542.          C  Shift surface points closer to nose by a decreasing
                  543.          C  interval between points
                  544.          C
ISN      149      545.                IF ((J .GT. 2) .AND. (PHI .EQ. 0.0)) THEN
ISN      150      546.                   DO 580 I=1,149
ISN      151      547.                      IF (THICK(151-I) .GT. 0.0) THEN
ISN      152      548.                         LAYRX(J,151-I) = LAYRX(J,151-I) -
                  549.               +                          ((0.4 * (20./DBLE(I+19))) *
                  550.               +                          DABS(LAYRX(J,151-I)-LAYRX(J,152-I)))
ISN      153      551.                         LAYRY(J,151-I) = LAYRY(J,151-I) +
                  552.               +                          ((0.4 * (20./DBLE(I+19))) *
                  553.               +                           DABS(LAYRY(J,151-I)-LAYRY(J,152-I)))
ISN      154      554.                      ENDIF
ISN      155      555.                      IF (THICK(I+151) .GT. 0.0) THEN
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN      156   556.                    LAYRX(J,I+151) = LAYRX(J,I+151) -
               557.       +                            ((0.4 * (20./DBLE(I+19))) *
               558.       +                            DABS(LAYRX(J,I+150)-LAYRX(J,I+151)))
ISN      157   559.                    LAYRY(J,I+151) = LAYRY(J,I+151) -
               560.       +                            ((0.4 * (20./DBLE(I+19))) *
               561.       +                            DABS(LAYRY(J,I+150)-LAYRY(J,I+151)))
ISN      158   562.                 ENDIF
ISN      159   563.           580  CONTINUE
ISN      160   564.              ENDIF
               565.     C
ISN      161   566.           300  CONTINUE
               567.     C
               568.     C  Write total mass accreted to table
               569.     C
ISN      162   570.              WRITE (8,590) TMASS
ISN      163   571.           590  FORMAT (8X,//,' Total mass accreted per metre: ',F9.5)
               572.     C
               573.     C  Call subroutine to plot airfoil and accreted layers
               574.     C
ISN      164   575.              CALL DRAW(ALPHM,BZERO,CHORD1,DROPDI,LAYERS,LWC,PHI,
               576.       +                SIMSPC,TAILEN,TAU,VEL)
               577.     C
ISN      165   578.           999  STOP
ISN      166   579.              END

*STATISTICS*   SOURCE STATEMENTS = 162, PROGRAM SIZE = 21362 BYTES, PROGRAM NAME = MAIN    PAGE:    1.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 1 ******
```

REQUESTED OPTIONS (PROCESS): SC(AXIS2,SYMBOL)

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(O) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                    *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                581.    C
                582.    ****************************************************************
                583.    *  SUBROUTINE DRAW - to plot airfoil  and all accretion       *
                584.    *                    layers, with appropriate information     *
                585.    *                    in the legend. This subroutine uses      *
                586.    *                    the plotting library *PLOTLIB            *
                587.    *                                                             *
                588.    *  Input Variables : as defined for Main program ;            *
                589.    *                    ALPHM,BZERO,CHORD1,DROPDI,LAYERS,LWC,    *
                590.    *                    PHI,SIMSPC,TAILEN,TAU,VEL                *
                591.    *                                                             *
                592.    *  Common Variables :LAYRX,LAYRY,PFOILX,PFOILY                *
                593.    *                                                             *
                594.    *  Other Variables : PLOTX,PLOTY - arrays containing x,y      *
                595.    *                    plotting data and scaling information    *
                596.    *                    for accretion layer currently being      *
                597.    *                    plotted                                  *
                598.    *                                                             *
                599.    ****************************************************************
                600.    C
ISN       1     601.          SUBROUTINE DRAW(ALPHM,BZERO,CHORD1,DROPDI,LAYERS,
                602.         +                LWC,PHI,SIMSPC,TAILEN,TAU,VEL)
                603.    C
ISN       2     604.          DOUBLE PRECISION PFOILX(302),PFOILY(302),LAYRX(20,300),
                605.         +                 LAYRY(20,300),PLOTX(302),PLOTY(302)
                606.    C
ISN       3     607.          DOUBLE PRECISION ALPHM,BZERO,CHORD,DROPDI,LWC,PHI,
                608.         +                 TAILEN,TAU,VEL
                609.    C
ISN       4     610.          INTEGER          LAYERS
                611.    C
ISN       5     612.          LOGICAL          SIMSPC
                613.    C
ISN       6     614.          COMMON /DR/LAYRX,LAYRY,PFOILX,PFOILY
                615.    C
                616.    C  To plot airfoil surface,
                617.    C  place scaling data at the end of coordinate arrays.
                618.    C
ISN       7     619.          PFOILX(301) = -0.4
ISN       8     620.          PFOILX(302) =  0.1
ISN       9     621.          PFOILY(301) = -0.25
ISN      10     622.          PFOILY(302) =  0.1
ISN      11     623.          PLOTX(301)  = -0.4
ISN      12     624.          PLOTX(302)  =  0.1
ISN      13     625.          PLOTY(301)  = -0.25
ISN      14     626.          PLOTY(302)  =  0.1
                627.    C
                628.    C  Open plotting library
                629.    C
ISN      15     630.          CALL PLOTS
ISN      16     631.          CALL METRIC(O)
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                632.    C
                633.    C  Draw 4 by 6 inch box
                634.    C
ISN      17     635.          CALL PLOT(2.0,2.0,-3)
ISN      18     636.          CALL PLOT(0.0,6.0,3)
ISN      19     637.          CALL PLOT(9.0,6.0,2)
ISN      20     638.          CALL PLOT(9.0,0.0,2)
                639.    C
                640.    C  Label axes and draw tics
                641.    C
ISN      21     642.          CALL AXIS2(0.0,0.0,'X/CHORD',-7,9.0,0.0,-0.4,0.1,1.0)
ISN      22     643.          CALL AXIS2(0.0,0.0,'Y/CHORD',7,6.0,90.0,-0.25,0.1,0.5)
                644.    C
                645.    C  Write legends to top left corner
                646.    C
ISN      23     647.          CALL SYMBOL(1.1,5.4,0.10,'DROPLET MVD  = ',0.0,15)
ISN      24     648.          CALL NUMBER(2.6,5.4,0.10,DROPDI,0.0,07)
ISN      25     649.          CALL SYMBOL(3.5,5.4,0.10,'M',0.0,01)
ISN      26     650.          CALL SYMBOL(1.1,5.2,0.10,'CHORD        = ',0.0,15)
ISN      27     651.          CALL NUMBER(2.6,5.2,0.10,CHORD1,0.0,04)
ISN      28     652.          CALL SYMBOL(3.5,5.2,0.10,'M',0.0,01)
ISN      29     653.          CALL SYMBOL(1.1,5.0,0.10,'AIR SPEED    = ',0.0,15)
ISN      30     654.          CALL NUMBER(2.6,5.0,0.10,VEL,0.0,02)
ISN      31     655.          CALL SYMBOL(3.5,5.0,0.10,'M / SEC',0.0,07)
ISN      32     656.          CALL SYMBOL(1.1,4.8,0.10,'LWC          = ',0.0,15)
ISN      33     657.          CALL NUMBER(2.6,4.8,0.10,LWC,0.0,05)
ISN      34     658.          CALL SYMBOL(3.5,4.8,0.10,'KG / M**3',0.0,09)
ISN      35     659.          CALL SYMBOL(1.1,4.6,0.10,'TIME STEP    = ',0.0,15)
ISN      36     660.          CALL NUMBER(2.6,4.6,0.10,TAU,0.0,02)
ISN      37     661.          CALL SYMBOL(3.5,4.6,0.10,'SEC',0.0,03)
ISN      38     662.          CALL SYMBOL(1.1,4.4,0.10,'BETA ZERO    = ',0.0,15)
ISN      39     663.          CALL NUMBER(2.6,4.4,0.10,BZERO,0.0,03)
ISN      40     664.          CALL SYMBOL(1.1,4.2,0.10,'ALPHA MAX    = ',0.0,15)
ISN      41     665.          CALL NUMBER(2.6,4.2,0.10,(ALPHM+TAILEN)*360./6.283,0.0,01)
ISN      42     666.          CALL SYMBOL(3.5,4.2,0.10,'DEGREES',0.0,07)
ISN      43     667.          CALL SYMBOL(1.1,4.0,0.10,'ATTACK ANGLE = ',0.0,15)
ISN      44     668.          CALL NUMBER(2.6,4.0,0.10,PHI*360./6.283,0.0,01)
ISN      45     669.          CALL SYMBOL(3.5,4.0,0.10,'DEGREES',0.0,07)
ISN      46     670.          CALL SYMBOL(1.1,3.8,0.10,'SPECTRUM:',0.0,09)
ISN      47     671.          IF ((SIMSPC) .OR. (PHI .NE. 0.0)) THEN
ISN      48     672.             CALL SYMBOL(2.6,3.8,0.10,'SIMULATED',0.0,09)
ISN      49     673.          ELSE
ISN      50     674.             CALL SYMBOL(2.6,3.8,0.10,'NO',0.0,02)
ISN      51     675.          ENDIF
                676.    C
                677.    C  Draw initial airfoil surface
                678.    C
ISN      52     679.          CALL LINE(PFOILX,PFOILY,300,2,0)
                680.    C
                681.    C  Transfer each layer in turn to the array holding
                682.    C  the scaling parameters
                683.    C
ISN      53     684.          DO 610 J = 2,LAYERS
ISN      54     685.             DO 700 L = 1,300
ISN      55     686.                PLOTX(L) =  LAYRX(J,L) / CHORD1
ISN      56     687.                PLOTY(L) =  LAYRY(J,L) / CHORD1
```

```
                        *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       57      688.      700     CONTINUE
                  689.    C
                  690.    C  Draw accretion layer
                  691.    C
ISN       58      692.            CALL LINE(PLOTX,PLOTY,300,2,0)
ISN       59      693.      610  CONTINUE
                  694.    C
                  695.    C  Close plotting library
                  696.    C
ISN       60      697.          CALL PLOT(1.0,1.0,999)
ISN       61      698.          RETURN
ISN       62      699.          END

*STATISTICS*   SOURCE STATEMENTS = 62, PROGRAM SIZE = 7298 BYTES, PROGRAM NAME = DRAW    PAGE:   12.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 2 ******
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:50      NAME: MAIN    PAGE:   15

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED          NOTEST   SEQ
                      OPTIMIZE(0)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                          *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*

                    700.     C ************************************************************
                    701.     C *  FUNCTION ONEBET - Calculates local collision efficiency*
                    702.     C *   estimate for a monodisperse droplet populations        *
                    703.     C *                                                          *
                    704.     C *                                                          *
                    705.     C *  Input Variables : as defined for main program;         *
                    706.     C *                       ALPHM,BZERO,ALPHA,SMDROP           *
                    707.     C *                                                          *
                    708.     C ************************************************************
                    709.     C
ISN         1       710.           DOUBLE PRECISION FUNCTION ONEBET(ALPHA,BZERO,ALPHM)
                    711.     C
ISN         2       712.           DOUBLE PRECISION BZERO,ALPHM,ALPHA
                    713.     C
                    714.     C
ISN         3       715.           IF (DABS(ALPHA) .GE. ALPHM) THEN
ISN         4       716.              ONEBET = 0.0
ISN         5       717.           ELSE
ISN         6       718.              ONEBET = DCOS(ALPHA) - (1.0 - BZERO)
ISN         7       719.           ENDIF
ISN         8       720.           RETURN
ISN         9       721.           END

*STATISTICS*   SOURCE STATEMENTS = 9, PROGRAM SIZE = 370 BYTES, PROGRAM NAME = ONEBET     PAGE:   15.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 3 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
          722.  C
          723.  C ***********************************************************
          724.  C *  Function SPCBET - Calculates the local c.e. for a      *
          725.  C *           simulated droplet size distribution           *
          726.  C *           scaled to the median volume droplet,          *
          727.  C *           and to (alpha max + tail). The tail           *
          728.  C *           length was determined in the main program.    *
          729.  C *                                                         *
          730.  C *  Input Variables : as defined for Main program;         *
          731.  C *                    ALPHA,ALPHM,BZERO,SMDROP,TAILEN      *
          732.  C *                                                         *
          733.  C *  Other Variables : X - scaled angle = Alpha /           *
          734.  C *                       (Alphm + Tailen)                  *
          735.  C *                                                         *
          736.  C ***********************************************************
          737.  C
ISN   1   738.        DOUBLE PRECISION FUNCTION SPCBET(ALPHA,BZERO,
          739.     +                                   ALPHM,TAILEN)
          740.  C
ISN   2   741.        DOUBLE PRECISION ALPHA,ALPHM,BZERO,TAILEN,X
          742.  C
ISN   3   743.        ALPHA = DABS(ALPHA)
ISN   4   744.        IF (ALPHA .GE. (ALPHM+TAILEN)) THEN
ISN   5   745.           SPCBET = 0.0
ISN   6   746.        ELSE
ISN   7   747.           X = ALPHA / (ALPHM+TAILEN)
ISN   8   748.              SPCBET = BZERO * (1.000 + 0.039 * X - 1.842 *
          749.     +              (X ** 2) - 0.543 * (X ** 3)
          750.     +              + 1.792 * (X **4) - 0.444 * (X ** 5))
ISN   9   751.              IF (SPCBET .LT. 0.0) SPCBET = 0.0
ISN  11   752.              IF (SPCBET .GT. BZERO) SPCBET = BZERO
ISN  13   753.        ENDIF
ISN  14   754.        RETURN
ISN  15   755.        END
```

*STATISTICS* SOURCE STATEMENTS = 13, PROGRAM SIZE = 608 BYTES, PROGRAM NAME = SPCBET PAGE: 16.

*STATISTICS* NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 4 ******

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
  756.    C
  757.    C ************************************************************
  758.    C  SUBROUTINE DROPS
  759.    C
  760.    C     This routine calculates a trajectory pair for water
  761.    C  droplets in a potential flow field around a NACA airfoil or
  762.    C  similar accretion shape. The stagnation line collision
  763.    C  efficiency is then determined.
  764.    C
  765.    C     The equation of motion of the droplet is integrated
  766.    C  by the Heun method (Mesinger and Arakawa, 1976: Numerical
  767.    C  Methods used in Atmospheric Models, Volume I. W.M.O. GARP
  768.    C  Publications Series No. 17). The time step is set to
  769.    C  1/400 of the time to travel one chord length at the free
  770.    C  stream speed. A maximum of 3000 steps are allowed,
  771.    C  by which time the droplet is assumed to have collided
  772.    C  with the airfoil or to have passed it.
  773.    C
  774.    C     The droplet's initial x-coordinate is set to an
  775.    C  upstream distance of 5.0 chord lengths.  At this
  776.    C  distance the x-speed is assumed to be equal to the air
  777.    C  speed in the x-direction, and the droplet y-speed is
  778.    C  set to zero.
  779.    C
  780.    C     A first trajectory is integrated beginning at y = zero,
  781.    C  which is then incremented by a small amount (DELTAY)
  782.    C  for the second trajectory. If both trajectories
  783.    C  intercept the airfoil, the local collision efficiency, and
  784.    C  the angle between the local surface normal and free stream
  785.    C  direction are calculated.
  786.    C
  787.    C     The formulation of Beard and Pruppacher (J.A.S., Vol. 26,
  788.    C  p. 1066, 1969) is used for the drag coefficient.  Potential
  789.    C  flow calculations are made according to the method of Kennedy
  790.    C  and Marsden (Can. Aero. and Space J., Vol. 22, p. 243, 1976)
  791.    C
  792.    C     One external subprogram is required, from the IMSL library:
  793.    C  the subroutine LEQTIF, to solve a system of equations
  794.    C  in the potential flow calculation.
  795.    C
  796.    C     The following internal subprograms are called from DROPS.
  797.    C
  798.    C  ******************** Subprograms **********************
  799.    C
  800.    C
  801.    C     SUBROUTINE COLLID -calculates final impact position
  802.    C       if the droplet collides with the surface.
  803.    C
  804.    C     SUBROUTINE FLOWXY- calculates the air speeds at a
  805.    C       given x,y location using the results of POFLO,
  806.    C       and droplet accelerations (from the equations of
  807.    C       motion ) for the current iteration and time step.
  808.    C
```

```
*....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      C     SUBROUTINE KCALC - calculates the matrix elements
      C       K used in POFLO and FLOWXY.
      C
      C     SUBROUTINE POFLO - calculates the potential vorticites
      C       at the surface of the airfoil according to the
      C       method of Kennedy and Marsden.
      C
      C     SUBROUTINE TRAJEC - calculates droplet trajectories
      C       by integrating the droplet equations of motion by
      C       the trapezoid rule with one iteration step.
      C
      C
      C     Input Variables:
      C
      C     CHORD      - airfoil chord length in metres
      C
      C     DROPDI     - droplet diameter in microns
      C
      C
      C     NPOINT     - number of points in the complete airfoil
      C                  surface (must be divisible by 15)
      C     VEL        - free stream air speed in metres per second
      C
      C     Output Variables :
      C
      C     BZERO      - stagnation line collision efficiency
      C
      C     Common Variables: (with Main)
      C
      C     GAMMA      - vector of potential vorticities for
      C                  potential flow calculation
      C
      C     XFOIL,     - vectors containing points which define
      C     YFOIL        the complete dimensioned airfoil surface
      C
      C
      C     Other Variables:
      C
      C     ALPHA      - angle between the local surface normal
      C                  and the free stream direction at
      C                  the impact point
      C
      C     ARC        - arc length along the airfoil surface
      C                  between the impact points of the two
      C                  trajectories (estimated by a straight
      C                  line segment)
      C     DELTAT     - time step in seconds
      C
      C     DELTAY     - difference in initial y position of
      C                  the two trajectories
      C
      C     FINX1,     - x-positions of the droplets at impact
      C     FINX2        for each trajectory
      C
      C     FINY1,     - y-positions of the droplets at impact
      C     FINY2        for each trajectory
```

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                865.    C
                866.    C     HALFPI     - pi / 2 (FORTRAN77 constant parameter)
                867.    C
                868.    C     HIT1,      - logical flags for each trajectory
                869.    C     HIT2         which indicate whether a collision
                870.    C                  with the airfoil has occurred
                871.    C
                872.    C     NSTEPS     - maximum number of time steps to be
                873.    C                  integrated
                874.    C
                875.    C     PI         - pi (FORTRAN77 constant parameter)
                876.    C
                877.    C     SLOPE1,    - slopes of the airfoil or accretion
                878.    C     SLOPE2       surface at the impact points
                879.    C
                880.    C     XINIT,     - initial droplet x,y position in metres
                881.    C     YINIT
                882.    C
                883.    C
                884.    C ***********************************************************
                885.    C
ISN      1      886.          SUBROUTINE DROPS(DROPDI,VEL,CHORD,NPOINT,BZERO)
                887.    C
ISN      2      888.          DOUBLE PRECISION XFOIL(600),YFOIL(600),GAMMA(41)
                889.    C
ISN      3      890.          DOUBLE PRECISION ARC,BZERO,CHORD,DELTAT,DELTAY,
                891.         +                 DROPDI,FINX1,FINX2,FINY1,FINY2,
                892.         +                 SLOPE1,SLOPE2,VEL,XINIT,YINIT
                893.    C
ISN      4      894.          INTEGER           NSTEPS,NPOINT
                895.    C
ISN      5      896.          LOGICAL           HIT1,HIT2
                897.    C
ISN      6      898.          PARAMETER (PI = 3.1415926535897932,HALFPI = PI / 2.0)
                899.    C
ISN      7      900.          COMMON /MN/ XFOIL,YFOIL,GAMMA
                901.    C
                902.    C
                903.    C  Set maximum number of time steps , initial x-coordinate, and
                904.    C  the difference in initial y-coordinates
                905.    C
ISN      8      906.              NSTEPS = 3000
ISN      9      907.              DELTAY = 1.0E-8
ISN     10      908.              XINIT  = 5.0 * CHORD
ISN     11      909.              YINIT  = 0.0
                910.    C
                911.    C  Calculate time step
                912.    C
ISN     12      913.              DELTAT = CHORD / (VEL * 400)
                914.    C
                915.    C  Calculate potential vorticity matrix for air speed calculations
                916.    C
ISN     13      917.              CALL POFLO(NPOINT,CHORD)
                918.    C
                919.    C  Integrate first trajectory
                920.    C
```

```
                    *...|*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       14   921.          CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CHORD,DELTAT,NSTEPS,
                922.         +             NPOINT,HIT1,FINX1,FINY1,SLOPE1)
                923.     C
                924.     C  Correct initial y position and integrate second trajectory
                925.     C
ISN       15    926.          YINIT = YINIT + DELTAY
ISN       16    927.          CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CHORD,DELTAT,NSTEPS,
                928.         +             NPOINT,HIT2,FINX2,FINY2,SLOPE2)
                929.     C
                930.     C  Test for result of trajectory (miss, graze or collision)
                931.     C
ISN       17    932.          IF ((HIT1) .AND. (HIT2)) THEN
                933.     C
                934.     C       Calculate impact parameters and local collision efficiency
                935.     C
ISN       18    936.            ARC    = DSQRT(((FINX2 - FINX1) ** 2) + ((FINY1 - FINY2)
                937.         +            ** 2))
ISN       19    938.            BZERO = DELTAY / ARC
                939.     C
                940.     C  If droplet didn't hit the foil, write message to screen
                941.     C
ISN       20    942.          ELSE IF ((.NOT. HIT1) .OR. (.NOT. HIT2)) THEN
ISN       21    943.            WRITE (6,40)
ISN       22    944.       40   FORMAT (//,'Droplet did not reach the airfoil.')
ISN       23    945.          ENDIF
                946.     C
ISN       24    947.          RETURN
ISN       25    948.          END

*STATISTICS*   SOURCE STATEMENTS = 25, PROGRAM SIZE = 926 BYTES, PROGRAM NAME = DROPS     PAGE:    17.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 5 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
*....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      C
      C ***************************************************************
      C *    SUBROUTINE TRAJEC                                        *
      C *                                                             *
      C *       Integrates trajectory of droplet whose initial        *
      C *    position is specified in the parameter list. Calls       *
      C *    POFLO to set up matrix for potential flow calculations,  *
      C *    FLOWXY for iterative calculations of air speed and       *
      C *    droplet acceleration, and COLLID to determine            *
      C *    impact points.                                           *
      C *                                                             *
      C *    INPUT VARIABLES:                                         *
      C *                  CHORD  - airfoil chord length              *
      C *                  DELTAT - time step                         *
      C *                  DROPDI - droplet diameter                  *
      C *                  NPOINT - number of surface points          *
      C *                  NSTEPS - maximum number of time steps      *
      C *                  VEL    - free stream air speed             *
      C *                  XINIT  - initial x position                *
      C *                  YINIT  - initial y position                *
      C *                                                             *
      C *    OUTPUT VARIABLES:                                        *
      C *                  FINALX - droplet x velocity on impact      *
      C *                  FINALY - droplet y velocity on impact      *
      C *                  HIT    - logical flag set to .TRUE. if     *
      C *                           a collision occurs                *
      C *                  SLOPE  - slope of the surface at impact    *
      C *                                                             *
      C *    COMMON VARIBLES:                                         *
      C *                  CO     - vector of cosines of angles of the*
      C *                           control elements                  *
      C *                  DELTA  - vector ot half-lengths of the     *
      C *                           control elements                  *
      C *                  GAMMA  - output array of POFLO, passed to  *
      C *                           FLOWXY                            *
      C *                  SI     - vector of sines of angles of the  *
      C *                           control elements                  *
      C *                  XCON,  - x,y coordinates of the control    *
      C *                  YCON     points                            *
      C *                  XFOIL, - vectors of initial surface        *
      C *                  YFOIL    coordinates                       *
      C *                                                             *
      C *    OTHER VARIABLES:                                         *
      C *                  DVX,   - arrays containing x,y components  *
      C *                  DVY      of droplet accelerations          *
      C *                  L      - counter for searching surface array*
      C *                  LINIT  - initial value of counter L        *
      C *                  VX,VY  - arrays containing x,y components  *
      C *                           of droplet speed for each step    *
      C *                  X,Y    - arrays containing x,y positions   *
      C *                           for each time step                *
      C *                                                             *
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      1002.    C*****************************************************************
      1003.    C
ISN        1  1004.          SUBROUTINE TRAJEC(XINIT,YINIT,DROPDI,VEL,CHORD,DELTAT,
      1005.         +                NSTEPS,NPOINT,HIT,FINALX,FINALY,SLOPE)
      1006.    C
ISN        2  1007.          DOUBLE PRECISION X(2500),Y(2500),GAMMA(41),XFOIL(600),
      1008.         +                 YFOIL(600),VX(2500),VY(2500),DVX(2500),
      1009.         +                 DVY(2500),DELTA(40),CO(40),SI(40),
      1010.         +                 XCON(41),YCON(41)
      1011.    C
ISN        3  1012.          DOUBLE PRECISION XINIT,YINIT,DROPDI,CHORD,VEL,DELTAT,
      1013.         +                 FINALX,FINALY,SLOPE
      1014.    C
ISN        4  1015.          INTEGER          NSTEPS,NPOINT,LINIT
      1016.    C
ISN        5  1017.          LOGICAL          HIT
      1018.    C
ISN        6  1019.          COMMON /MN/ XFOIL,YFOIL,GAMMA/DRP/DELTA,CO,SI,XCON,YCON
      1020.    C
      1021.    C  Initialize values for integration
      1022.    C
ISN        7  1023.              X(1)  = -XINIT
ISN        8  1024.              Y(1)  = YINIT
ISN        9  1025.              VY(1) = 0.0
ISN       10  1026.              VX(1) = VEL
ISN       11  1027.              I     = 0
ISN       12  1028.              HIT   = .FALSE.
      1029.    C
      1030.    C  Begin integration
      1031.    C
ISN       13  1032.        100   I = I+1
ISN       14  1033.              IF (I .GE. NSTEPS) GOTO 250
      1034.    C
      1035.    C  Calculate current step air speeds and accelerations
      1036.    C
ISN       15  1037.                 CALL FLOWXY(X(I),Y(I),VX(I),VY(I),VEL,CHORD,DROPDI,
      1038.         +                    NPOINT,DVX(I),DVY(I))
      1039.    C
      1040.    C  Calculate next step droplet speeds and positions
      1041.    C
ISN       16  1042.                 VX(I+1) = VX(I) - (DELTAT * DVX(I))
ISN       17  1043.                 VY(I+1) = VY(I) + (DELTAT * DVY(I))
ISN       18  1044.                 X(I+1)  = X(I) + (VX(I) * DELTAT) - (0.5 * DVX(I)
      1045.         +                    * (DELTAT * DELTAT))
ISN       19  1046.                 Y(I+1)  = Y(I) + (VY(I) * DELTAT) + (0.5 * DVY(I)
      1047.         +                    * (DELTAT * DELTAT))
      1048.    C
      1049.    C  Calculate next step air speeds and accelerations
      1050.    C
ISN       20  1051.                 CALL FLOWXY(X(I+1),Y(I+1),VX(I+1),VY(I+1),VEL,
      1052.         +                    CHORD,DROPDI,NPOINT,DVX(I+1),DVY(I+1))
      1053.    C
      1054.    C  Iterate calculations of next step droplet speed and
      1055.    C  position using averaged values of accelerations
      1056.    C
ISN       21  1057.                 VX(I+1) = VX(I) - ((DELTAT / 2.0) * (DVX(I) +
```

```
                                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                    1058.           +                DVX(I+1)))
ISN         22      1059.                VY(I+1) = VY(I) + ((DELTAT / 2.0) * (DVY(I) +
                    1060.           +                DVY(I+1)))
ISN         23      1061.                X(I+1) =  X(I) + (VX(I) * DELTAT) - (0.25 *
                    1062.           +                (DVX(I) + DVX(I+1)) * (DELTAT * DELTAT))
ISN         24      1063.                Y(I+1) =  Y(I) + (VY(I) * DELTAT) + (0.25 *
                    1064.           +                (DVY(I) + DVY(I+1)) * (DELTAT * DELTAT))
                    1065.     C
                    1066.     C  If trajectory has reached the nose
                    1067.     C  test whether trajectory has intercepted the airfoil surface
                    1068.     C
                    1069.     C  Set LINIT to begin search on upper or lower surface
                    1070.     C  if the droplet has reached to within one current chord
                    1071.     C  length of the nose
                    1072.     C
ISN         25      1073.                LINIT = 0
ISN         26      1074.                L = LINIT
ISN         27      1075.           200     CONTINUE
ISN         28      1076.                   L = L + 1
ISN         29      1077.                   IF (L .GE. (LINIT + 125)) GOTO 100
                    1078.     C
                    1079.     C  Search for adjacent pair of points which bracket the
                    1080.     C  y-coordinate of current trajectory position (if pair
                    1081.     C  cannot be found, continue with next integration step)
                    1082.     C
ISN         30      1083.                   IF ((YFOIL(L+1) .GT. Y(I+1)) .AND. (YFOIL(L) .LT.
                    1084.           +          Y(I+1))) THEN
                    1085.     C
                    1086.     C  Check if current position has passed the airfoil surface
                    1087.     C  between the two points
                    1088.     C
ISN         31      1089.                      IF ((XFOIL(L+1) .LT. X(I+1)) .OR. (XFOIL(L) .LT.
                    1090.           +             X(I+1))) THEN
ISN         32      1091.                         HIT    = .TRUE.
                    1092.     C
                    1093.     C  Determine the impact point on the surface, and local
                    1094.     C  surface slope at that point , then exit from subroutine
                    1095.     C
ISN         33      1096.                         CALL COLLID(X(I),X(I+1),Y(I),Y(I+1),XFOIL(L),
                    1097.           +                         YFOIL(L),XFOIL(L+1),YFOIL(L+1),
                    1098.           +                         FINALX,FINALY)
ISN         34      1099.                         SLOPE = (YFOIL(L+1) - YFOIL(L)) / DABS(XFOIL
                    1100.           +                        (L+1) - XFOIL(L))
ISN         35      1101.                         GOTO 250
ISN         36      1102.                      ELSE
                    1103.     C
                    1104.     C  If not inside surface, continue integration
                    1105.     C
ISN         37      1106.                         GOTO 100
ISN         38      1107.                      ENDIF
ISN         39      1108.                   ELSE
                    1109.     C
                    1110.     C  Else continue search for bracketing pair
                    1111.     C
ISN         40      1112.                      GOTO 200
ISN         41      1113.                   ENDIF
```

```
                      *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       42     1114.      250     CONTINUE
                 1115.    C
ISN       43     1116.           RETURN
ISN       44     1117.           END
```

*STATISTICS* SOURCE STATEMENTS = 44, PROGRAM SIZE = 122570 BYTES, PROGRAM NAME = TRAJEC PAGE: 21.

*STATISTICS* NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 6 ******

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      1118.     C ***************************************************************
      1119.     C *                                                             *
      1120.     C *  SUBROUTINE POFLO - Uses the method of Kennedy and          *
      1121.     C *    Marsden (ref) to solve for the matrix of potential       *
      1122.     C *    vorticites about the airfoil surface.  A reduced         *
      1123.     C *    number of points is used to define the surface for       *
      1124.     C *    this subroutine than for the accretion section.          *
      1125.     C *    Subroutine KCALC is called to calculate the matrix       *
      1126.     C *    of K values.                                             *
      1127.     C *                                                             *
      1128.     C *  INPUT VARIABLES  :                                         *
      1129.     C *                      CHORD  - airfoil chord length          *
      1130.     C *                      NPOINT - number of points defining     *
      1131.     C *                               airfoil surface               *
      1132.     C *  COMMON VARIABLES :                                         *
      1133.     C *                      CO,SI  - vectors of cosines and sines  *
      1134.     C *                               of the control element angles*
      1135.     C *                      DELTA  - vector containing control     *
      1136.     C *                               element half lengths          *
      1137.     C *                      GAMMA  - vector containing the         *
      1138.     C *                               potential vorticities on      *
      1139.     C *                               the airfoil surface           *
      1140.     C *                      XCON,  - vectors containing control    *
      1141.     C *                      YCON     point coordinates             *
      1142.     C *                      XFOIL, - vectors of surface            *
      1143.     C *                      YFOIL    point coordinates             *
      1144.     C *  OTHER VARIABLES :                                          *
      1145.     C *                      IDGT,    routine                       *
      1146.     C *                      IER                                    *
      1147.     C *                      K      - vector input to matrix        *
      1148.     C *                               solver subroutine LEQT1F      *
      1149.     C *                      N      - NPOINT/15, the no. of surface*
      1150.     C *                               points used to calculate      *
      1151.     C *                               the stream function           *
      1152.     C *                      WKAREA,- input required for LEQT1F     *
      1153.     C *                      XF,YF  - reduced vectors of surface    *
      1154.     C *                               points (length NPOINT/15)     *
      1155.     C *                                                             *
      1156.     C ***************************************************************
      1157.     C
ISN      1    1158.           SUBROUTINE POFLO(NPOINT,CHORD)
      1159.     C
ISN      2    1160.           DOUBLE PRECISION GAMMA(41),XFOIL(600),YFOIL(600),
      1161.          +                  XCON(41),YCON(41),XF(41),YF(41),
      1162.          +                  K(41,41),WKAREA(41),DELTA(40),
      1163.          +                  CO(40),SI(40),CHORD
      1164.     C
ISN      3    1165.           INTEGER NPOINT,N,IDGT,IER
      1166.     C
ISN      4    1167.           COMMON /MN/ XFOIL,YFOIL,GAMMA/DRP/DELTA,CO,SI,XCON,YCON
      1168.     C
      1169.     C  Construct new surface from every 15th point of old surface
      1170.     C  non-dimensionalised to the current chord length
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                 1171.    C
ISN         5    1172.          N = NPOINT / 15
ISN         6    1173.          XF(1)   = XFOIL(1) / CHORD
ISN         7    1174.          YF(1)   = YFOIL(1) / CHORD
ISN         8    1175.          XF(N+1) = XF(1)
ISN         9    1176.          YF(N+1) = YF(1)
ISN        10    1177.          DO 10 I=2, N
ISN        11    1178.             XF(I) = XFOIL((15 * (I-1)) + 1) / CHORD
ISN        12    1179.             YF(I) = YFOIL((15 * (I-1)) + 1) / CHORD
ISN        13    1180.       10  CONTINUE
                 1181.    C
                 1182.    C  Form arrays of control points
                 1183.    C
ISN        14    1184.          DO 20 I = 1,N
ISN        15    1185.             XCON(I) = (XF(I) + XF(I+1)) / 2.0
ISN        16    1186.             YCON(I) = (YF(I) + YF(I+1)) / 2.0
ISN        17    1187.       20  CONTINUE
                 1188.    C
                 1189.    C  Define the trailing point
                 1190.    C
ISN        18    1191.          XCON(N+1) = ((XF((N/2)+1) - XCON(N/2)) * 0.01) + XF((N/2)+1)
ISN        19    1192.          YCON(N+1) = 0.0
                 1193.    C
                 1194.    C  Calculate R (as defined in Kennedy and Marsden) for each
                 1195.    C  control point, store in GAMMA
                 1196.    C
ISN        20    1197.          DO 30 J = 1,N+1
ISN        21    1198.             GAMMA(J) = (YCON(J))
                 1199.    C
                 1200.    C  Set last column of K(I,J) to 1.0
                 1201.    C
ISN        22    1202.             IF (J .EQ. N+1) THEN
ISN        23    1203.                DO 35 I=1,N+1
ISN        24    1204.                   K(I,N+1) = 1.0
ISN        25    1205.       35       CONTINUE
                 1206.    C
                 1207.    C  For all other J, calculate the control element half lengths
                 1208.    C  and functions of the angle to the x-axis.
                 1209.    C
ISN        26    1210.             ELSE
ISN        27    1211.                DELTA(J) = DSQRT(((XF(J+1) - XF(J)) ** 2) +
                 1212.         +                   ((YF(J+1) - YF(J)) ** 2)) / 2.0
ISN        28    1213.                CO(J)    = (XF(J+1) - XF(J)) / (2.0 * DELTA(J))
ISN        29    1214.                SI(J)    = (YF(J+1) - YF(J)) / (2.0 * DELTA(J))
                 1215.    C
                 1216.    C  Calculate K for all remaining I,J
                 1217.    C
ISN        30    1218.                DO 40 I=1,N
ISN        31    1219.                   CALL KCALC(XCON(J),YCON(J),XCON(I),YCON(I),
                 1220.         +                    DELTA(J),CO(J),SI(J),K(I,J))
ISN        32    1221.       40       CONTINUE
ISN        33    1222.             ENDIF
ISN        34    1223.       30  CONTINUE
                 1224.    C
                 1225.    C  Call IMSLDPLIB routine to solve matrix equation;
                 1226.    C  output matrix of potential vorticites is in GAMMA.
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:52      NAME: POFLO   PAGE:   27

                        *....*...1.....2.....3.....4.....5.....6.....7:*.....8.....9.....*

                1227.     C
ISN       35    1228.               IDGT = 8
ISN       36    1229.               CALL LEQT1F(K,1,N+1,N+1,GAMMA,IDGT,WKAREA,IER)
ISN       37    1230.               WRITE (6,*) GAMMA
                1231.     C
ISN       38    1232.               RETURN
ISN       39    1233.               END

*STATISTICS*   SOURCE STATEMENTS = 39, PROGRAM SIZE = 16358 BYTES, PROGRAM NAME = POFLO     PAGE:    25.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 7 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
1234.    C ***************************************************************
1235.    C *                                                             *
1236.    C *  SUBROUTINE FLOWXY  -  calculates for the current step:     *
1237.    C *      air speed components from matrix equation for          *
1238.    C *      potential flow streamline , using the output           *
1239.    C *      matrix GAMMA from POFLO, and KCALC to generate the     *
1240.    C *      matrix elements K. Droplet accelerations are then      *
1241.    C *      calculated according to the equations of motion        *
1242.    C *      from the air speed, droplet Reynolds number, and       *
1243.    C *      drag coefficient according to the formulation of       *
1244.    C *      Beard and Pruppacher.                                  *
1245.    C *                                                             *
1246.    C *  INPUT VARIABLES :                                         *
1247.    C *                CHORD  - chord length                        *
1248.    C *                DROPDI - droplet diameter                    *
1249.    C *                NPOINT - no of airfoil surface points        *
1250.    C *                VEL    - free stream air speed               *
1251.    C *                VX,VY, - current droplet speeds              *
1252.    C *                X,Y    - current droplet positions           *
1253.    C *                                                             *
1254.    C *  OUTPUT VARIABLES :                                        *
1255.    C *                DVX,DVY - current droplet accelerations      *
1256.    C *                                                             *
1257.    C *  COMMON VARIABLES :                                        *
1258.    C *                CO,SI,GAMMA,DELTA,XCON,YCON,XFOIL,YFOIL      *
1259.    C *                                                             *
1260.    C *  OTHER VARIABLES :                                         *
1261.    C *                COEFF   - combined cefficients of the        *
1262.    C *                          equations of motion                *
1263.    C *                DRAGCO  - drag coefficient                   *
1264.    C *                DROPRE  - droplet Reynolds number            *
1265.    C *                GRIDX,  - gridpoints used in the finite      *
1266.    C *                GRIDY     difference calculation of air      *
1267.    C *                          speed from the stream function     *
1268.    C *                N       - NPOINT/15, the no. of surface      *
1269.    C *                          points used to calculate the       *
1270.    C *                          stream function                    *
1271.    C *                K, R    - variables  for the calculation     *
1272.    C *                          of stream function                 *
1273.    C *                PSI     - stream function of the air flow    *
1274.    C *                          field                              *
1275.    C *                SUMK    - array containing the sum of K      *
1276.    C *                          times GAMMA(J) for I gridpoints    *
1277.    C *                UX,UY   - current air speed components       *
1278.    C *                                                             *
1279.    C ***************************************************************
1280.    C
ISN     1  1281.          SUBROUTINE FLOWXY(X,Y,VX,VY,VEL,CHORD,DROPDI,NPOINT,
1282.         +                      DVX,DVY)
1283.    C
ISN     2  1284.          DOUBLE PRECISION CO(40),DELTA(40),GAMMA(41),GRIDX(4),
1285.         +                 GRIDY(4),PSI(4),SI(40),SUMK(4),
1286.         +                 XCON(41),YCON(41),XFOIL(600),
```

```
*....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
                 1287.       +                  YFOIL(600)
                 1288.    C
ISN         3    1289.          DOUBLE PRECISION COEFF,CHORD,DRAGCO,DROPDI,DROPRE,DVX,
                 1290.       +                  DVY,K,R,UX,UY,VEL,VX,VY,X,Y
                 1291.    C
ISN         4    1292.          INTEGER          N,NPOINT
                 1293.    C
ISN         5    1294.          COMMON /MN/XFOIL,YFOIL,GAMMA/DRP/DELTA,CO,SI,XCON,YCON
                 1295.    C
                 1296.    C  Set up non-dimensional grid for finite difference calculation
                 1297.    C  of the streamfunction derivatives
                 1298.    C
ISN         6    1299.          GRIDX(1) = X / CHORD
ISN         7    1300.          GRIDX(2) = X / CHORD
ISN         8    1301.          GRIDX(3) = (X - (DROPDI / 2.0)) / CHORD
ISN         9    1302.          GRIDX(4) = (X + (DROPDI / 2.0)) / CHORD
ISN        10    1303.          GRIDY(1) = (Y + (DROPDI / 2.0)) / CHORD
ISN        11    1304.          GRIDY(2) = (Y - (DROPDI / 2.0)) / CHORD
ISN        12    1305.          GRIDY(3) = Y / CHORD
ISN        13    1306.          GRIDY(4) = Y / CHORD
                 1307.    C
                 1308.    C  Define N as in POFLO
                 1309.    C
ISN        14    1310.          N = NPOINT / 15
                 1311.    C
                 1312.    C  Calculate for each grid point, R and the sum of products
                 1313.    C  K times GAMMA(J), for J =1, N
                 1314.    C
ISN        15    1315.          DO 10 I =1,4
ISN        16    1316.             R = (GRIDY(I))
ISN        17    1317.             SUMK(I) = 0.0
ISN        18    1318.             DO 20 J = 1,N
ISN        19    1319.                CALL KCALC(XCON(J),YCON(J),GRIDX(I),GRIDY(I),
                 1320.       +                   DELTA(J),CO(J),SI(J),K)
ISN        20    1321.                SUMK(I) = SUMK(I) + (GAMMA(J) * K)
ISN        21    1322.          20    CONTINUE
                 1323.    C
                 1324.    C  Calculate stream functions
                 1325.    C
ISN        22    1326.                PSI(I) = -SUMK(I) + R - GAMMA(N+1)
ISN        23    1327.          10  CONTINUE
                 1328.    C
                 1329.    C  Calculate dimensional air speeds at point x,y from finite
                 1330.    C  difference derivatives of the stream functions
                 1331.    C
ISN        24    1332.          UX = VEL * CHORD * (PSI(1) - PSI(2)) / DROPDI
ISN        25    1333.          UY = VEL * CHORD * (PSI(3) - PSI(4)) / DROPDI
                 1334.    C
                 1335.    C Calculate Reynolds number and drag coefficient
                 1336.    C
ISN        26    1337.          DROPRE    = DROPDI * DSQRT(((VX - UX) ** 2) + ((VY - UY)
                 1338.       +              ** 2)) / 1.262E-5
ISN        27    1339.          IF (DROPRE .LT. 2.0) THEN
ISN        28    1340.             DRAGCO = (24.0 / DROPRE) + (2.45 / (DROPRE ** 0.045))
ISN        29    1341.          ELSE IF (DROPRE .LT. 21.0) THEN
ISN        30    1342.             DRAGCO = (24.0 / DROPRE) + (2.76 / (DROPRE ** 0.198))
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       31    1343.           ELSE IF (DROPRE .GT. 200.0) THEN
ISN       32    1344.              DRAGCO = (24.0 / DROPRE) + (4.73 / (DROPRE ** 0.37))
                1345.         +             + (6.24E-3 * (DROPRE ** 0.38))
ISN       33    1346.           ELSE
ISN       34    1347.              DRAGCO = (24.0 / DROPRE) + (4.536 / (DROPRE ** 0.368))
ISN       35    1348.           ENDIF
                1349.     C
                1350.     C  Calculate accelerations according to the equations of motion
                1351.     C
ISN       36    1352.           COEFF = 9.9225E-4 * (DRAGCO / DROPDI)
ISN       37    1353.           DVX   = COEFF * DSQRT(((VX - UX) **2 ) + ((VY - UY) ** 2))
                1354.         +         * (VX - UX)
ISN       38    1355.           DVY   = -COEFF * DSQRT(((VX - UX) ** 2) + ((VY - UY) ** 2))
                1356.         +         * (VY - UY)
                1357.     C
ISN       39    1358.           RETURN
ISN       40    1359.           END

*STATISTICS*   SOURCE STATEMENTS = 40, PROGRAM SIZE = 2150 BYTES, PROGRAM NAME = FLOWXY    PAGE:   28.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 8 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      1360.     C ****************************************************************
      1361.     C *                                                              *
      1362.     C *  SUBROUTINE KCALC - calculates a single element of the       *
      1363.     C *                     matrix K for the control points I        *
      1364.     C *                     and J                                    *
      1365.     C *                                                              *
      1366.     C *  INPUT DATA :                                                *
      1367.     C *                  CO,SI    - direction angles of the segment  *
      1368.     C *                  DELTA    - half length of the control       *
      1369.     C *                             element J                        *
      1370.     C *                  XCI,XCJ  - x-coordinates of control points  *
      1371.     C *                             I and J , or of grid point I     *
      1372.     C *                             and control point J              *
      1373.     C *                  YCI,YCJ  - y-coordinates of above           *
      1374.     C *                                                              *
      1375.     C *  OUTPUT VARIBLES :                                           *
      1376.     C *                  K        - matrix element                   *
      1377.     C *                                                              *
      1378.     C *  OTHER VARIABLES :                                           *
      1379.     C *                  A,B      - geometrical quantities defined   *
      1380.     C *                             in Figure 2, Kennedy and Marsden *
      1381.     C *                  R1SQ,    - squares of R1 and R2, also       *
      1382.     C *                  R2SQ       defined in Figure 2 of K&M       *
      1383.     C *                  TERM1,   - individual terms of the          *
      1384.     C *                  TERM2,     expression for K                 *
      1385.     C *                  TERM3,                                      *
      1386.     C *                  DENOM                                       *
      1387.     C *                                                              *
      1388.     C ****************************************************************
      1389.     C
ISN        1  1390.           SUBROUTINE KCALC(XCJ,YCJ,XCI,YCI,DELTA,CO,SI,K)
      1391.     C
ISN        2  1392.           DOUBLE PRECISION A,B,CO,DELTA,DENOM,K,R1SQ,R2SQ,
      1393.          +                    SI,TERM1,TERM2,TERM3,XCI,XCJ,YCI,YCJ
      1394.     C
ISN        3  1395.           PARAMETER (PI = 3.1415926535897932)
      1396.     C
      1397.     C  Calculate K
      1398.     C
ISN        4  1399.           B  = ((XCI-XCJ) * CO) + ((YCI-YCJ) * SI)
ISN        5  1400.           A  = ((YCI-YCJ) * CO) - ((XCI-XCJ) * SI)
ISN        6  1401.           R1SQ = (A*A) + ((B+DELTA)*(B+DELTA))
ISN        7  1402.           R2SQ = (A*A) + ((B-DELTA)*(B-DELTA))
ISN        8  1403.           TERM1 = (B + DELTA) * DLOG(R1SQ)
ISN        9  1404.           TERM2 = (B - DELTA) * DLOG(R2SQ)
ISN       10  1405.           DENOM = (A*A) + (B*B) - (DELTA*DELTA)
ISN       11  1406.           IF (DENOM .LE. 1.0D-30) THEN
ISN       12  1407.              IF (DABS(A) .LT. 1.0D-30) THEN
ISN       13  1408.                 TERM3 = 2.0 * A * PI
ISN       14  1409.              ELSE
ISN       15  1410.                 TERM3 = 2.0 * A * (DATAN((B + DELTA) / A) -
      1411.          +                    DATAN((B - DELTA) / A))
ISN       16  1412.              ENDIF
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN       17    1413.            ELSE
ISN       18    1414.               TERM3 = 2.0 * A * DATAN(2.0 * A * DELTA / DENOM)
ISN       19    1415.            ENDIF
                1416.    C
ISN       20    1417.            K = (TERM1 - TERM2 + TERM3 - (4.0 * DELTA)) / (4.0 * PI)
                1418.    C
ISN       21    1419.            RETURN
ISN       22    1420.            END

*STATISTICS*   SOURCE STATEMENTS = 22, PROGRAM SIZE = 1060 BYTES, PROGRAM NAME = KCALC    PAGE:   31.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 9 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOUNT(60)

```
                 *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
      1421.      C ***************************************************************
      1422.      C *                                                             *
      1423.      C *  SUBROUTINE COLLID   -  Calculates the x,y position         *
      1424.      C *    of impact on the airfoil surface                         *
      1425.      C *                                                             *
      1426.      C *  INPUT VARIABLES:                                           *
      1427.      C *                XOUT,   - last droplet position outside      *
      1428.      C *                YOUT      the airfoil                        *
      1429.      C *                XIN,    - first droplet position inside      *
      1430.      C *                YIN       the airfoil                        *
      1431.      C *                XF1,XF2 - x and y coordinates of the end-    *
      1432.      C *                YF1,YF2   points of the surface segment      *
      1433.      C *                          intercepted by the trajectory      *
      1434.      C *  OUTPUT VARIABLES :                                         *
      1435.      C *                FINALX, - estimated coordinates of the       *
      1436.      C *                FINALY    impact point on the surface        *
      1437.      C *                                                           *
      1438.      C *  OTHER VARIABLES :                                          *
      1439.      C *                SLOPTR  - slope of the trajectory segment    *
      1440.      C *                          which crosses the airfoil surface *
      1441.      C *                SLOPSF  - slope of the surface segment       *
      1442.      C *                          intercepted by the trajectory      *
      1443.      C *                INT1,   - intercepts for the linear          *
      1444.      C *                INT2      equations of the two segments      *
      1445.      C *                                                           *
      1446.      C *                                                           *
      1447.      C ***************************************************************
      1448.      C
ISN       1     1449.            SUBROUTINE COLLID (XOUT,XIN,YOUT,YIN,XF1,YF1,XF2,YF2,
      1450.           +                       FINALX,FINALY)
      1451.      C
ISN       2     1452.            DOUBLE PRECISION XOUT,XIN,YOUT,YIN,XF1,XF2,YF1,YF2,
      1453.           +                    SLOPTR,SLOPSF,INT1,INT2,FINALX,FINALY
      1454.      C
      1455.      C  Calculate slope of trajectory at impact point
      1456.      C
ISN       3     1457.                SLOPTR = (YIN - YOUT) / (XIN - XOUT)
      1458.      C
      1459.      C  Calculate slope of surface segment at impact point
      1460.      C
ISN       4     1461.                SLOPSF = (YF2 - YF1) / (XF2 - XF1)
      1462.      C
      1463.      C  Find intercepts of the two linear equations
      1464.      C
ISN       5     1465.                INT1 = YIN - (SLOPTR * XIN)
ISN       6     1466.                INT2 = YF2 - (SLOPSF * XF2)
      1467.      C
      1468.      C  Find the impact point as the intersection point
      1469.      C  of the two segments
      1470.      C
ISN       7     1471.                FINALX = (INT2 - INT1) / (SLOPTR - SLOPSF)
ISN       8     1472.                FINALY = (SLOPTR * FINALX) + INT1
      1473.      C
```

```
                    *....*...1.....2.....3.....4.....5.....6.....7.*.....8.....9.....*
ISN          9      1474.          RETURN
ISN         10      1475.          END

*STATISTICS*   SOURCE STATEMENTS = 10, PROGRAM SIZE = 512 BYTES, PROGRAM NAME = COLLID     PAGE:   33.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 10 ******
```

```
LEVEL 1.1.1 (DEC  81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:54      NAME: MAIN    PAGE:   35
SUMMARY OF MESSAGES AND STATISTICS FOR ALL COMPILATIONS
*STATISTICS*   SOURCE STATEMENTS = 162, PROGRAM SIZE = 21362 BYTES, PROGRAM NAME = MAIN    PAGE:    1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
*STATISTICS*   SOURCE STATEMENTS = 62, PROGRAM SIZE = 7298 BYTES, PROGRAM NAME = DRAW    PAGE:   12.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 2 ******
*STATISTICS*   SOURCE STATEMENTS = 9, PROGRAM SIZE = 370 BYTES, PROGRAM NAME = ONEBET    PAGE:   15.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 3 ******
*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 608 BYTES, PROGRAM NAME = SPCBET    PAGE:   16.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 4 ******
*STATISTICS*   SOURCE STATEMENTS = 25, PROGRAM SIZE = 926 BYTES, PROGRAM NAME = DROPS    PAGE:   17.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 5 ******
*STATISTICS*   SOURCE STATEMENTS = 44, PROGRAM SIZE = 122570 BYTES, PROGRAM NAME = TRAJEC    PAGE:   21.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 6 ******
*STATISTICS*   SOURCE STATEMENTS = 39, PROGRAM SIZE = 16358 BYTES, PROGRAM NAME = POFLO    PAGE:   25.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 7 ******
*STATISTICS*   SOURCE STATEMENTS = 40, PROGRAM SIZE = 2150 BYTES, PROGRAM NAME = FLOWXY    PAGE:   28.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 8 ******
*STATISTICS*   SOURCE STATEMENTS = 22, PROGRAM SIZE = 1060 BYTES, PROGRAM NAME = KCALC    PAGE:   31.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 9 ******
*STATISTICS*   SOURCE STATEMENTS = 10, PROGRAM SIZE = 512 BYTES, PROGRAM NAME = COLLID    PAGE:   33.
```

```
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 10 ******
******* SUMMARY STATISTICS ******* 0 DIAGNOSTICS GENERATED. HIGHEST SEVERITY CODE IS 0.
```

## Appendix VI

### The MVD Approximation for Collision Efficiency

The task of computing volume-weighted averages of collision efficiencies for a droplet size spectrum is often simplified by substituting a monodisperse size spectrum at the median volume diameter of the distribution. As discussed in Chapter III, this can lead, in some cases, to a significant underestimation of the maximum accretion angle. However, it has often been assumed that the weighted overall and stagnation line collision efficiencies are well approximated by the mvd values. In this Appendix the validity of that assumption is examined.

Table A.1 lists a variety of atmospheric and marine droplet size spectra, as relative numbers of droplets in 10 or 3μm wide size bins. These spectra were either measured in the field, or in icing wind tunnels, or estimated from empirical parameterized distributions, and cover a range of types and mvds.

They are derived from the following sources: Bain and Gayet (1984), Battan and Reitan (1957), Best (1951), Khrgian-Mazin, as reported by Pruppacher and Klett (1980), McComber and Touzot (1981), Preobrazhenskii (1973), and Squires (1958). Those marked FROST were measured by the oil

| mid-band droplet diameter | relative number of drops per unit volume | | | | | | |
|---|---|---|---|---|---|---|---|
| | Khrgian-Mazin 1 | Khrgian-Mazin 2 | Khrgian- azin 3 | Khrgian-Mazin 4 | Squires Hawaiian | Squires dark stratus | Squires Trade-wind cu |
| 5μm | 0.724 | 0.069 | 0.014 | 0.002 | 0.25 | 6.5 | 9 |
| 15 | 2.397 | 0.340 | 0.082 | 0.012 | 0.65 | 6.0 | 16 |
| 25 | 2.449 | 0.518 | 0.148 | 0.024 | 1.0 | 5.4 | 29 |
| 35 | 1.766 | 0.557 | 0.189 | 0.035 | 0.8 | 5.1 | 21 |
| 45 | 1.074 | 0.505 | 0.203 | 0.043 | 0.8 | 2.5 | 3.5 |
| 55 | 0.590 | 0.414 | 0.198 | 0.047 | 0.7 | 0.5 | |
| 65 | 0.303 | 0.317 | 0.180 | 0.049 | 0.35 | 0.2 | |
| 75 | 0.149 | 0.232 | 0.156 | 0.048 | 0.15 | 0.1 | |
| 85 | 0.070 | 0.164 | 0.131 | 0.046 | 0.10 | 0.05 | |
| 95 | 0.032 | 0.112 | 0.106 | 0.042 | 0.06 | | |
| 105 | 0.014 | 0.075 | 0.085 | 0.038 | 0.05 | | |
| 115 | 0.006 | 0.049 | 0.066 | 0.034 | 0.04 | | |
| 125 | 0.003 | 0.032 | 0.051 | 0.030 | 0.03 | | |
| 135 | 0.001 | 0.021 | 0.039 | 0.026 | 0.04 | | |
| 145 | | 0.013 | 0.029 | 0.022 | 0.01 | | |
| 155 | | 0.008 | 0.022 | 0.019 | | | |
| 165 | | 0.005 | 0.016 | 0.016 | | | |
| 175 | | 0.003 | 0.012 | 0.013 | | | |
| 185 | | 0.002 | 0.009 | 0.011 | | | |
| 195 | | 0.001 | 0.006 | 0.009 | | | |
| 205 | | | 0.004 | 0.007 | | | |
| 215 | | | 0.003 | 0.006 | | | |
| 225 | | | | 0.005 | | | |
| 235 | | | | 0.004 | | | |
| 245 | | | | 0.003 | | | |

Table A.1 - Droplet size spectra from many different sources (see text for references). -

| mid-band droplet diameter | relative number of drops per unit volume | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Battan and Reitan fairweather cumulus | Battan and Reitan stratus | FROST 1 | FROST 2 | FROST 3 | FROST 4 | FROST 5 | Preobrazhenskii |
| 5μm | 600 | 100 | 127 | 94 | 97 | 3 | 17 | 5180 |
| 15 | 70 | 15 | 327 | 167 | 213 | 37 | 68 | 1620 |
| 25 | 13 | 1.1 | 104 | 33 | 86 | 63 | 64 | 480 |
| 35 | 0.26 | 0.2 | 13 | 6 | 33 | 58 | 31 | 200 |
| 45 | | | 1 | 4 | 14 | 51 | 12 | 110 |
| 55 | | | | | 2 | 35 | 11 | 56 |
| 65 | | | | | 1 | 26 | 9 | 32 |
| 75 | | | | | | 17 | 3 | 21 |
| 85 | | | | | | 14 | 2 | 14 |
| 95 | | | | | | 16 | 4 | 10 |
| 105 | | | | | | 12 | 9 | 7 |
| 115 | | | | | | 5 | 2 | 5 |
| 125 | | | | | | 4 | 3 | 4 |
| 135 | | | | | | 8 | 2 | 3 |
| 145 | | | | | | 7 | 1 | 2 |
| 155 | | | | | | 2 | 0 | 1 |
| 165 | | | | | | 1 | 2 | |
| 175 | | | | | | 2 | | |
| 185 | | | | | | 2 | | |

Table A.1 - continued.

relative number of drops per unit volume

| mid-band droplet diameter | McComber and Touzot | mid-band droplet diameter | Bain and Gayet 1 | Bain and Gayet 2 | Bain and Gayet 3 |
|---|---|---|---|---|---|
| 5.1µm | 945 | 1.5 | 20 | 7 | 13.0 |
| 15.3 | 1093 | 4.5 | 41 | 12 | 15.5 |
| 25.5 | 481 | 7.5 | 30 | 23 | 15.5 |
| 35.7 | 211 | 10.5 | 7.5 | 33 | 18.0 |
| 45.9 | 74 | 13.5 | 2.5 | 18 | 15.5 |
| 56.1 | 38 | 16.5 | | 5 | 12.5 |
| 66.3 | 30 | 19.5 | | 2 | 6.0 |
| 76.5 | 13 | 22.5 | | | 2 |
| 86.7 | 19 | 25.5 | | | 2 |
| 96.9 | 11 | | | | |
| 107.1 | 10 | | | | |
| 117.3 | 3 | | | | |
| 127.5 | 1 | | | | |
| 137.7 | 1 | | | | |

percent volume

| mid-band droplet diameter | Best 1 | Best 2 | Best 3 | Best 4 | Best 5 |
|---|---|---|---|---|---|
| 5µm | 63.2 | 22 | 4 | 3 | 36 |
| 15 | 36.8 | 41 | 24 | 31 | 47 |
| 25 | | 26 | 39 | 53 | 15 |
| 35 | | 9 | 26 | 13 | 2 |
| 45 | | 2 | 6 | | |
| 55 | | | 1 | | |

**Table A.1 - continued.**

slide technique in the FROST icing wind-tunnel facility at The University of Alberta.

For each spectrum, the mvd was calculated following the method derived by Lozowski (1978). Assuming a uniform distribution of droplets within each bin, the partial volume of each is given by:

$$V_i = (n_i/24)\ \pi\ w^3\ (i^4 - (i-1)^4), \qquad \text{A.8}$$

where $n_i$ is the number of drops in the ith bin, and w is the bin width, say 10 μm. Then, if V is the total volume of all droplets, let

$$u_i = (\Sigma V_i)\ /\ V. \qquad \text{A.9}$$

Now find $u_k$ such that $u_{k-1} < 0.5$, and $u_k > 0.5$. Finally,

$$\text{mvd} = w\ (((0.5 - u_{k-1})\ /\ (u_k - u_{k-1}))\ (k^4 - (k-1)^4) + (k-1)^4)^{0.25}, \qquad \text{A.10}$$

Overall and stagnation line collision efficiencies have been calculated for the mvd of each spectrum. For the complete spectrum, weighted average collision efficiencies are calculated using the fractional volume of each bin as the weighting factor for the collision efficiency of the mvd of that bin.

All of the collision efficiencies are calculated according to Table II.6, and for the same conditions, i.e. U = 10 m/sec and $D_c$ = 0.0254 m. The results are shown in Table A.2 and Figures A.2 and A.3. For a different type of comparison, the conditions have been varied for a single spectrum (labelled as Bain and Gayet 2 in Table A.1) to give a complete range of collision efficiencies. These results are compared in Table A.3 and in Figures A.4 and A.5.

Also calculated are collision efficiencies for the droplet at the 42nd volume percentile (d42), on a suggestion made by Lozowski and Makkonen (private communication, 1986), based on a theoretical analysis using Gaussian quadrature to estimate the overall collision efficiency for a droplet spectrum. These data are included in the figures and tables listed above.

Comparison of Figure A.4 with Figure 1 of Makkonen (1984), which shows the same type of calculations for the same spectrum used here as well as two other spectra given by Bain and Gayet, 1982), reveals a major disagreement in the sign of $E_{spec} - E_{mvd}$. The disagreement is due to a misinterpretation of the original droplet spectrum diagram in Bain and Gayet (1982) on the part of Makkonen (private communication, 1986). Figure A.4 reflects the correct interpretation.

| Spectrum | $\beta_{spec}$ | $E_{spec}$ | mvd | $\beta_{mvd}$ | $E_{mvd}$ | d42 | $\beta_{d42}$ | $E_{d42}$ |
|---|---|---|---|---|---|---|---|---|
| Khrgian-Mazin 1 | 0.824 | 0.710 | 56.9μm | 0.852 | 0.745 | 52.1μm | 0.835 | 0.718 |
| Khrgian-Mazin 2 | 0.906 | 0.836 | 93.6 | 0.924 | 0.862 | 86.1 | 0.914 | 0.846 |
| Khrgian-Mazin 3 | 0.936 | 0.886 | 126.2 | 0.949 | 0.906 | 116.5 | 0.943 | 0.895 |
| Khrgian-Mazin 4 | 0.955 | 0.919 | 165.9 | 0.964 | 0.933 | 154.5 | 0.960 | 0.927 |
| Squires Hawaiian | 0.886 | 0.804 | 81.4 | 0.908 | 0.835 | 70.9 | 0.889 | 0.804 |
| Squires dark stratus | 0.767 | 0.626 | 43.0 | 0.790 | 0.650 | 39.7 | 0.769 | 0.619 |
| Squires Trade-wind Cu | 0.702 | 0.532 | 34.9 | 0.730 | 0.565 | 33.2 | 0.714 | 0.543 |
| Battan and Reitan Cu | 0.393 | 0.224 | 18.0 | 0.459 | 0.249 | 16.6 | 0.419 | 0.211 |
| Battan and Reitan stratus | 0.364 | 0.199 | 17.2 | 0.434 | 0.225 | 15.9 | 0.396 | 0.191 |
| FROST 1 | 0.555 | 0.362 | 24.7 | 0.603 | 0.402 | 22.6 | 0.565 | 0.357 |
| FROST 2 | 0.583 | 0.402 | 26.2 | 0.628 | 0.432 | 23.4 | 0.580 | 0.375 |
| FROST 3 | 0.680 | 0.516 | 35.0 | 0.731 | 0.566 | 31.2 | 0.694 | 0.515 |
| FROST 4 | 0.928 | 0.873 | 127.7 | 0.949 | 0.907 | 111.3 | 0.939 | 0.889 |
| FROST 5 | 0.919 | 0.859 | 112.3 | 0.940 | 0.890 | 107.1 | 0.936 | 0.883 |
| Preobrazhenskii | 0.838 | 0.745 | 78.5 | 0.903 | 0.827 | 67.9 | 0.883 | 0.793 |
| McComber and Touzot | 0.836 | 0.736 | 75.9 | 0.899 | 0.820 | 65.8 | 0.878 | 0.786 |
| Bain and Gayet 1 | 0.108 | 0.036 | 8.8 | 0.088 | 0.010 | 8.4 | 0.065 | 0.003 |
| Bain and Gayet 2 | 0.284 | 0.123 | 13.3 | 0.301 | 0.118 | 12.5 | 0.267 | 0.096 |
| Bain and Gayet 3 | 0.421 | 0.232 | 17.5 | 0.444 | 0.233 | 16.6 | 0.418 | 0.210 |
| Best 1 | 0.415 | 0.247 | 18.3 | 0.467 | 0.255 | 17.0 | 0.429 | 0.220 |
| Best 2 | 0.582 | 0.401 | 26.9 | 0.639 | 0.445 | 25.0 | 0.610 | 0.409 |
| Best 3 | 0.542 | 0.350 | 24.4 | 0.599 | 0.397 | 22.5 | 0.564 | 0.357 |
| Best 4 | 0.313 | 0.160 | 15.3 | 0.375 | 0.173 | 13.1 | 0.291 | 0.111 |
| Best 5 | 0.186 | 0.070 | 9.4 | 0.120 | 0.021 | 9.0 | 0.098 | 0.014 |
| standard error | | | | 0.031 | 0.029 | | 0.027 | 0.026 |

Table A.2 – Overall and stagnation line collision efficiencies as plotted in Figures A.2 and A.3.

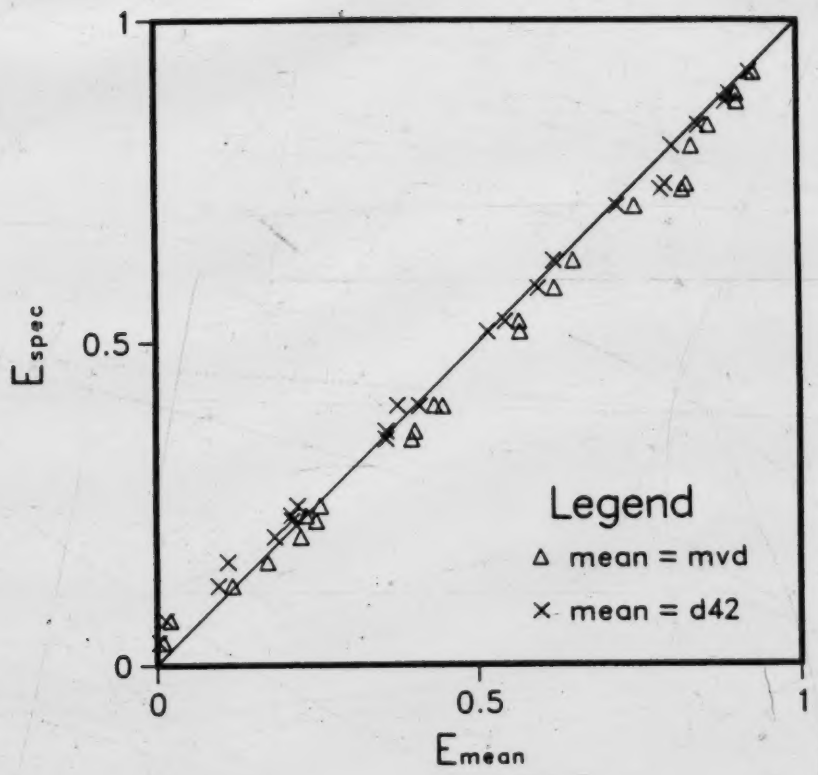



Figure A.2 - Overall collision efficiencies calculated for many different drop size spectra vs. E's calculated for the mvd and d42 of each spectrum. In each case U=10 m/sec, $D_c$=0.0254 m.

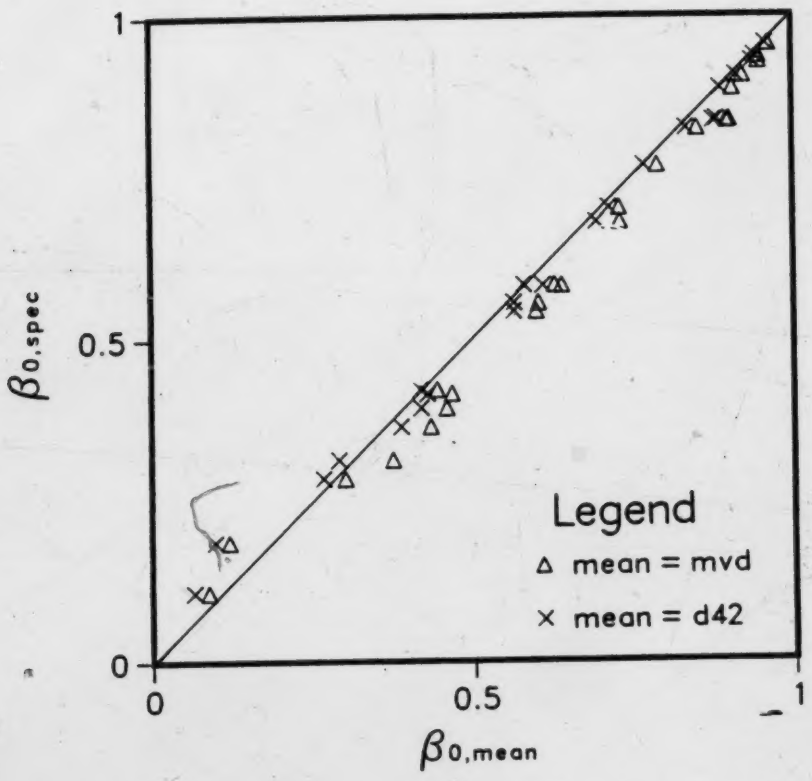



Figure A.3 - Stagnation line collision efficiencies calculated for many different drop size spectra vs. $\beta_0$'s calculated for the mvd and d42 of each spectrum. In each case U=10 m/sec, $D_c$=0.0254 m.

| U | $\beta_{o_{spec}}$ | $E_{spec}$ | $\beta_{o_{mvd}}$ | $E_{mvd}$ | $\beta_{o_{d42}}$ | $E_{d42}$ |
|---|---|---|---|---|---|---|
| 5m/sec | 0.121 | 0.037 | 0.121 | 0.022 | 0.089 | 0.010 |
| 7 | 0.202 | 0.072 | 0.208 | 0.062 | 0.175 | 0.044 |
| 10 | 0.284 | 0.123 | 0.301 | 0.118 | 0.267 | 0.096 |
| 15 | 0.383 | 0.194 | 0.403 | 0.196 | 0.370 | 0.169 |
| 20 | 0.448 | 0.250 | 0.470 | 0.257 | 0.439 | 0.228 |
| 25 | 0.496 | 0.295 | 0.518 | 0.305 | 0.489 | 0.275 |
| 30 | 0.532 | 0.331 | 0.555 | 0.344 | 0.527 | 0.314 |
| 40 | 0.586 | 0.389 | 0.608 | 0.405 | 0.582 | 0.374 |
| 50 | 0.624 | 0.432 | 0.646 | 0.450 | 0.622 | 0.420 |
| 80 | 0.693 | 0.516 | 0.713 | 0.536 | 0.692 | 0.508 |
| 200 | 0.790 | 0.648 | 0.806 | 0.668 | 0.791 | 0.646 |
| 350 | 0.830 | 0.708 | 0.844 | 0.726 | 0.832 | 0.708 |
| 700 | 0.865 | 0.765 | 0.877 | 0.781 | 0.867 | 0.766 |
| 1500 | 0.892 | 0.811 | 0.901 | 0.825 | 0.894 | 0.812 |
| standard error | | | 0.012 | 0.013 | 0.011 | 0.011 |

Table A.3 - Overall and stagnation line collision efficiencies as plotted in Figures A.4 and A.5.

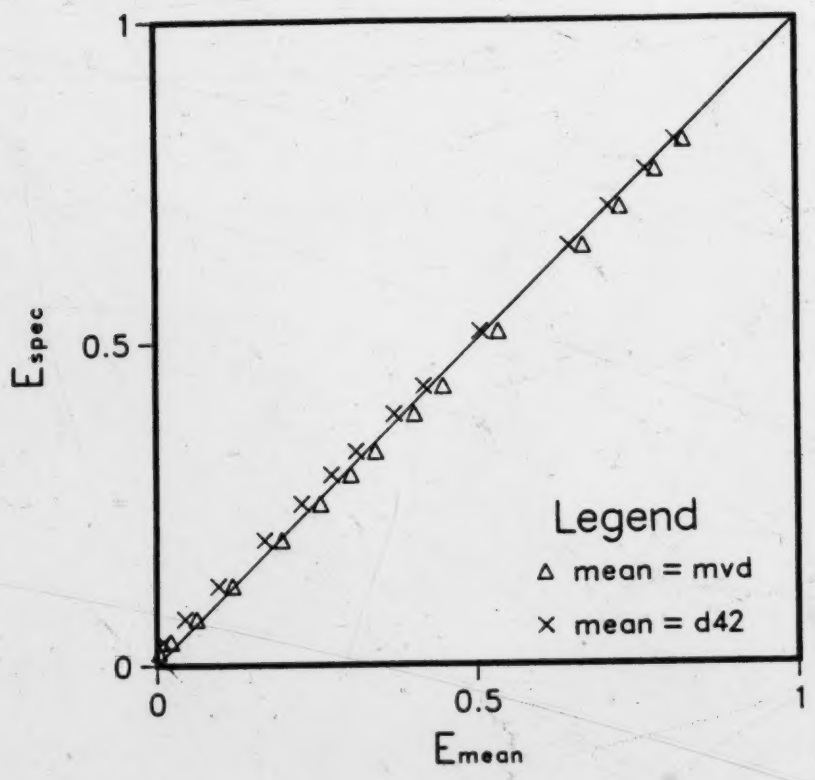



Figure A.4 - Overall collison efficiencies calculated for a single drop size spectrum (Bain and Gayet 2) at many different wind speeds, vs. E's calculated for the mvd and d42 of the same spectrum for the same wind speeds. $D_c$ = 0.0254 m.

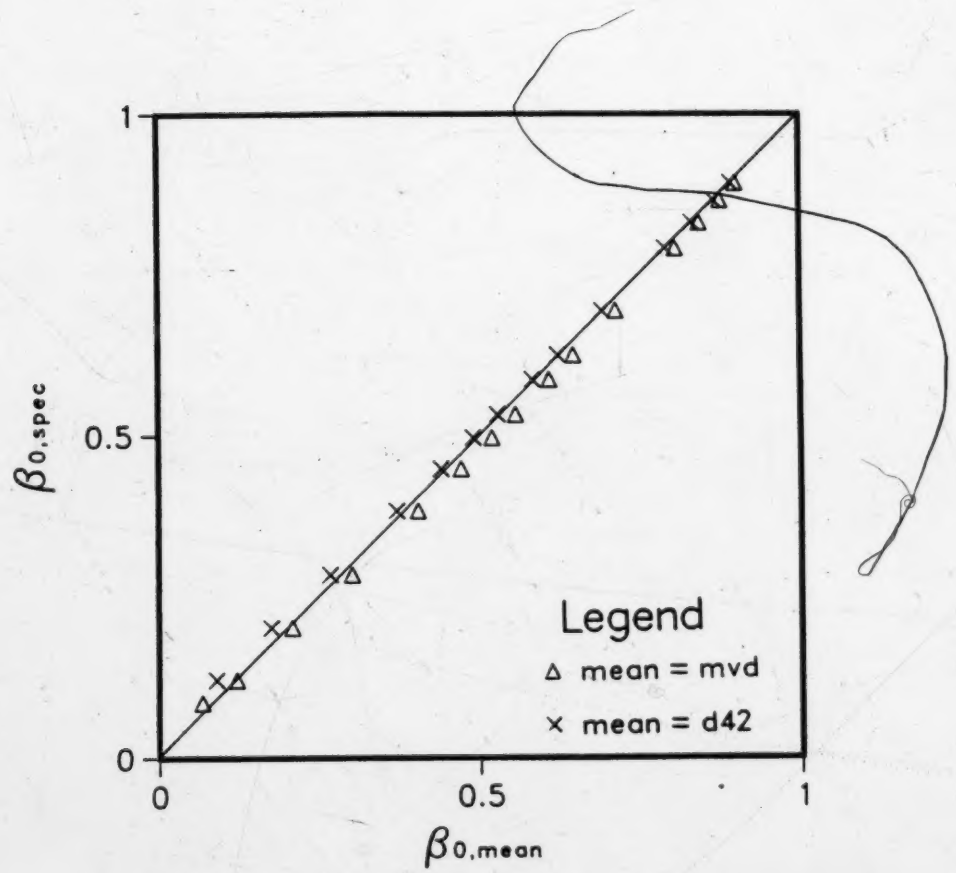



Figure A.5 - Stagnation line collision efficiencies calculated for a single drop size spectrum (Bain and Gayet 2) at many different wind speeds, vs. $\beta_0$'s calculated for the mvd and d42 of the same spectrum for the same wind speeds. $D_c$ = 0.0254 m.

The conclusion which may be made from this exercise is that, although the standard errors are not significantly different over the entire range of collision efficiencies (see Tables A.2 and A.3), in the range of E or $\beta_0$ less than about 0.2 the mvd approximation is preferable. Above that range, for most types of spectra, the d42 approximation gives better agreement with the spectrum-weighted collision efficiencies. However, for certain types of spectra which depart significantly from a Gaussian distribution (e.g. the double peaked spectrum from McComber and Touzot) even the d42 value may be in error by more than 10 percent.

## References


Bain, M. and Gayet, J-F. 1982: Contribution to the Modelling of the Ice Accretion Process: Ice Density Variation with the Impacted Surface Angle. Proceedings, First International Workshop on Atmospheric Icing of Structures, Hanover, 13-20.

Battan, L.J. and Reitan, C.H. 1957: Droplet Size Measurements in Convective Clouds. Artificial Stimulation of Rain, Weickmann and Smith, ed.s. London, Pergamon Press, p.184.

Best, A.C. 1951: Drop-size Distribution in Cloud and Fog. *Quart. Jour. Roy. Meteor. Soc.* 77, 418-426.

Lozowski, E.P. 1978: Stochastic Effects in Spray Droplet Sampling with Oiled Slides. National Research Council, Division of Mechanical Engineering, Laboratory Memorandum LT-172, 17 pp.

Makkonen, L. 1984: Modeling of Ice Accretion on Wires. *J. Climate and App. Meteor.*, 23, 929-939.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Circular Cylinder by the Finite Element Method. *J. Atmos.*

*Sci.* 38, 1027-1036.

Preobrazhenskii, L.Yu. 1973: Estimate of the Content of Spray Drops in the Near-Water Layer of the Atmosphere. *Fluid Mechanics - Soviet Research*, 2, 95-100.

Pruppacher, H.R. and Klett, J.D. 1980: Microphysics of Clouds and Precipitation. D. Reidel , p.11.

Squires, P. 1958: The Microstructure and Colloidal Stability of Warm Clouds: I. The Relation between Structure and Stability. *Tellus*, 10, , 256-261.